Package ‘tidyMicro’

March 28, 2020

Title A Pipeline for Microbiome Analysis and Visualization

Version 1.43

Date 2020-03-17

Maintainer Charlie Carpenter <charles.carpenter@cuanschutz.edu>

Description A reliable alternative to popular microbiome analysis R packages. We provide standard tools as well as novel extensions on standard analyses to improve interpretability and the analyst’s ability to communicate results, all while maintaining object malleability to encourage open source collaboration.

Depends R (>= 3.5.0), tidyverse (>= 1.3.0)

Imports magrittr (>= 1.5.0), ggrepel (>= 0.8.1), MASS (>= 7.3-51.4), VGAM (>= 1.1-2), rlang (>= 0.3.4), car (>= 3.0-3), lme4 (>= 1.1-21), vegan (>= 2.5-5), Matrix (>= 1.2-17), cowplot (>= 0.9.4), lsr (>= 0.5), shapes (>= 1.2.4), Evomorph (>= 0.9), ThreeWay (>= 1.1.3), factoextra (>= 1.0.5), ade4 (>= 1.7-13), scatterplot3d (>= 0.3-41), gridExtra (>= 2.3), plotly (>= 4.9.0), png (>= 0.1-7), latex2exp(>= 0.4.0), broom (>= 0.5.0), plyr (>= 1.8.0), dplyr (>= 0.8.0), ggplot2 (>= 3.2.0), purrr (>= 0.3.0), stringr (>= 1.4.0), tibble (>= 2.1.0), tidyr (>= 1.0.0), scales (>= 1.1.0)

Suggests knitr, markdown, roxygen2, rmarkdown

Encoding UTF-8

License GPL-3

LazyData true

RoxygenNote 7.1.0

BugReports https://github.com/CharlieCarpenter/tidyMicro

VignetteBuilder knitr

NeedsCompilation no

Author Charlie Carpenter [aut, cre],
Dan Frank [aut],
Kayla Williamson [aut],
Rachel Johnson [ctb]

Repository CRAN

Date/Publication 2020-03-28 14:20:02 UTC

R topics documented:

`alpha_div` *Alpha Diversity Calculations for tidy_micro*

Description

A wrapper function to calculate Sobs, Choa1, Goods, Shannon's diversity and evenness, and Simpson's diversity and evenness alpha diversities for your micro_set. Estimates are calculated based on rarefied bootstrapped samples

Usage

```
alpha_div(micro_set, table = NULL, iter = 100, min_depth = 0, min_goods = 0)
```

Arguments

`micro_set`	A tidy_micro data set
`table`	OTU table of interest
`iter`	The number of bootstrap resamples used for estimation
`min_depth`	Filter out libraries with sequencing depth (Total) below min_depth
`min_goods`	Filter out libraries Good's coverage below min_goods

Details

If you have multiple otu tables, you can specify the table you'd like to use to calculate your alpha diversities using the `table` option. We highly recommend using the lowest taxonomic rank available to calculate your alpha diversity. If you would like to calculate alpha diversities for each otu table in your micro_set, you can leave the `table` option as `NULL` and the function will calculate the alpha diversity for each table. The function will append the estimated alpha diversities to the tidy_micro supplied. The alpha diversity columns will be just before your clinical data. Since alpha diversity is estimated for each individual library (Lib), it will be repeated within each taxa block.

Value

A tidy_micro set with alpha diversity columns added in to the left of clinical data

Note

Be aware of your minimal sequencing depth as this will be the size of all bootstrapped resamples (rarefied).

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)
otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)

set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week


## calculate alpha diversity for every table
set_alpha <- set %>% alpha_div(min_depth = 5000, min_goods = 90)

## calculate alpha diversity for a specific table
set_fam_alpha <- set %>% alpha_div(table = "Family", min_depth = 5000, min_goods = 90)
```

`bb_bars` *Create stacked bar charts based on beta binomial model estimates*

Description

bb_bars takes the output from bb_mods and creates stacked bar charts of the estimated relative abundance for each taxa. The benefit of modeling each taxa before created stacked bar charts is the ability to control for potential confounders. The function will facet wrap interaction terms. Currently, only quant_style = "discrete" can be used for an interaction between two quantitative variables

Usage

```
bb_bars(
  modsum,
  ...,
  range,
  quant_style = c("continuous", "discrete"),
  top_taxa = 0,
  RA = 0,
  specific_taxa,
  lines = TRUE,
  xaxis,
  main,
  subtitle,
  xlab,
  ylab = "Relative Abundance (%)",
  facet_labels,
  facet_layout = 1
)
```

Arguments

`modsum` The output from bb_mods

`...` The covariate you'd like to plot. Can be an interaction term or main effect, but must be in the models created by bb_mods

`range` The range you'd like to plot over for a quantitative variable. Will default to the first and third quartiles

`quant_style` "continuous" will plot over the entire range specified; "discrete" will plot only the endpoints of the range specified. "continuous" by default. This option is ignored without a quantitative variable

`top_taxa` Only plot X taxa with the highest relative abundance. The rest will be aggregated into an "Other" category

`RA` Only plot taxa with a relative abundance higher than X. The rest will be aggregated into an "Other" category `specific_taxa` Character; Plot these specific taxa even if it doesn't meet the top_taxa or RA requirements

`lines` Logical; Add outlines around the different taxa colors in the stacked bar charts

`xaxis` Labels for the x-axis ticks. Most useful for categorical variables and defaults to the levels of the variable

`main` Plot title

`subtitle` Subtitle for the plot

`xlab` x-axis label

`ylab` y-axis label

`facet_labels` Labels for the facets created for interaction terms

`facet_layout` Rearrange the facets created for interaction terms

Value

Returns a ggplot that you can add geoms to if you'd like

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)
otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Creating beta binomial models on filtered tidy_micro set

bb_phy <- set %>%
otu_filter(ra_cutoff = 0.1, exclude_taxa = c("Unclassified", "Bacteria")) %>%
bb_mods(table = "Phylum", bpd1)

bb_phy %>%
bb_bars(bpd1, top_taxa = 4, xlab = "BPD Severity")
```

`bb_mods` *Fit beta binomial models to each taxa within an OTU table*

Description

Fit beta binomial models to each taxa within an OTU table through `vglm` in the **VGAM** package. Summaries for models or confidence intervals that fail to converge will not be returned, but taxa summaries will be provided in the output. Rank-Sum tests or presence/absence tests can be run on these taxa using `tidi_rank_sum` or `tidi_chisq`, respectively

Usage

```
bb_mods(
  micro_set,
  table,
  ...,
  CI_method = c("wald", "profile"),
  SS_type = c(2, 3, "II", "III"),
  trace = FALSE
)
```

Arguments

`micro_set`	A tidy_micro data set
`table`	OTU table of interest
`...`	Covariates of interest. Can be interactions such as Group*Age
`CI_method`	Character indicating the type of method used for confidence interval estimation. Wald intervals are the current default. Abbreviations allowed. See `confintvglm` for more details
`SS_type`	Type of sums of squares calculated in `anova.vglm`. Either type II (2) or type III (3) sums of squares. Type II is the default
`trace`	Print messages of model fitting proceedure

Details

Models containing only fixed effects are fit using `vglm` in the **VGAM** package. ANOVA / ANCOVA tests are conducted using a Likelihood Ratio test

Value

A list containing several different model components and summaries

`Convergend_Summary`	A data.frame of model summaries from convergent models. Includes the Taxa name, the model coefficient, the estimated beta, the beta's 95 percent confidence interval, Z score, p_value, false discovery rate p-value, and p-value from likelihood ratio test
`Estimate_Summary`	A data.frame of model estimates from convergent models intended to be ready for export for publications. Includes the Taxa name, the model coefficient, the estimated Rate Ratio, the Wald 95 percent confidence interval, the Z-score, and false discovery rate p-value
`RA_Summary`	A data.frame of taxa summaries. Includes the Taxa name, grouping variables (each factor variable in your models), sample size (n), percent of 0 counts, basic summaries of relative abundance, percentiles of relative abundance, and a logical indicator of whether or not the model converged
`formula`	The formula used in the model
`Model_Coef`	Model coefficients (used in plotting funcitons)
`Model_Covs`	Model covariates (used in plotting functions)

Note

False Discovery Rate p-values are calculated using p.adjust. Estimated rate ratios and confidence intervals for interactions in the Estimate_Summary table include all main effects. It is not simply the exponentiated interaction beta, it is the interaction of the sum of the intercept, corresponding main effect betas, and interaction betas

References

anova.vglm, vglm, betabinomial

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs <- list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
  filter(day == 7) ## Only including first week


bb_phy <- set %>%

## Filtering out low abundance and unclassified taxa
## These models will either break or we don't care about them
otu_filter(prev_cutoff = 5, ra_cutoff = 0.1,
           exclude_taxa = c("Unclassified", "Bacteria")) %>%

## Beta binomial models for each Family of taxa with bpd1 as a covariate
bb_mods(table = "Phylum", bpd1, CI_method = "wald")

names(bb_phy)
bb_phy$Estimate_Summary
```

beta_div *Beta Diversity Calculations for tidy_micro*

Description

Calculate beta diversities of your tidy_micro set. This function reformats the data into the original OTU table and then feeds that into the vegdist function

Usage

```
beta_div(micro_set, table, method = "bray")
```

Arguments

`micro_set`	A tidy_micro data set
`table`	Table you'd like to use when calculating alpha diversity. Your lowest level is recommended
`method`	A dissimilarity method compatible with `vegdist`

Value

A symmetrix distance matrix

References

`vegdist`

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Bray-Curtis beta diversity
bray <- set %>% beta_div(table = "Family")

## Morisita-Horn beta diversity
horn <- set %>% beta_div(table = "Family", method = "horn")
```

`beta_heatmap` *Create heatmaps of the supplied dissimilarity matrices*

Description

Create heatmaps of the supplied dissimilarity matrices ordered by supplied grouping variables

Usage

```
beta_heatmap(
  beta_div,
  micro_set,
  ...,
  low_grad,
  high_grad,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  subtitle = NULL,
  natural_order = TRUE,
  legend_title = "Dissimilarity"
)
```

Arguments

`beta_div`	A dissimilarity matrix calculated by `beta_div`
`micro_set`	A tidy_micro data set
`...`	Variables for ordering
`low_grad`	Colors for the corelation magnitude. Will be fed into scale_fill_gradient
`high_grad`	Colors for the corelation magnitude. Will be fed into scale_fill_gradient
`main`	Plot title
`xlab`	x-axis label
`ylab`	y-axis label
`subtitle`	Plot label
`natural_order`	Keep order of axes in the conventional order for dissimilarity matrices
`legend_title`	Title for the legend

Value

Returns a ggplot that you can add geoms to if you'd like

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Bray-Curtis beta diversity
bray <- set %>% beta_div(table = "Family")

bray %>% beta_heatmap(micro_set = set, bpd1)
```

`cla` *An OTU table of class level taxa counts*

Description

Infants who required mechanical ventilation had tracheal aspirates samples collected at 7, 14, and 21 days of age (+/- 48 hours). Infants who were mechanically ventilated and had at least one tracheal aspirate collected were included in this study. Subjects were required to be enrolled within 7 days of age. Bacterial profiles were determined by broad-range amplification and sequence analysis of 16S rRNA genes.

Usage

```
cla
```

Format

A 34x75 data.frame

`OTU_Name` A character vector of class level OTU names

`Lib names` The following columns are the sequencing counts for each library with library names

Source

https://doi.org/10.1371/journal.pone.0170120

`clin` *A data set containing the clinical data of the subjects sequenced*

Description

Infants who required mechanical ventilation had tracheal aspirates samples collected at 7, 14, and 21 days of age (+/- 48 hours). Infants who were mechanically ventilated and had at least one tracheal aspirate collected were included in this study. Subjects were required to be enrolled within 7 days of age. Bacterial profiles were determined by broad-range amplification and sequence analysis of 16S rRNA genes

Usage

```
clin
```

Format

A 74x8 data.frame

`study_id` A character vector of study IDs

`weight` A numeric vector of infant birth weights (Kg)

`sex` A factor; infant sex

`gestational_age` A numeric vector of gestational age in weeks

`mom_ethncty` A factor; maternal ethnicity

`bpd1` A factor; BPD severity

`day` A numeric vector; days of life at time of sequencing

`Lib` A character vector of sequencing library names

Source

https://doi.org/10.1371/journal.pone.0170120

cor_heatmap *Create correlation heatmaps of taxa and another continuous variable*

Description

Calculated the correlation between a specified continuous variable and some taxa measure. Correlation type and taxa measure (count, relative abundance, etc.) can be specified by the user but is "spearman" and relative abundance, respectively, by default

Usage

```
cor_heatmap(
  micro_set,
  table,
  ...,
  y = clr,
  method = c("pearson", "kendall", "spearman"),
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  subtitle = NULL,
  legend_title = NULL,
  low_grad,
  high_grad
)
```

Arguments

`micro_set`	A tidy_micro data set
`table`	The OTU table
`...`	Continuous variables of interest
`y`	The taxa information: cts, ra, etc. The centered log ratio (clr) is recommended.
`method`	Correlation type; must be supported by `cor`. By default it is "spearman" to use with clr. If you'd like to use taxa ra, it is recommend you switch to Kendall's correlation to account for the large number of ties common in taxa ra (lots of 0s)
`main`	Plot title
`xlab`	x-axis label
`ylab`	y-axis label
`subtitle`	Plot label
`legend_title`	Title for the legend
`low_grad`	Colors for the corelation magnitude. Will be fed into scale_fill_gradient
`high_grad`	Colors for the corelation magnitude. Will be fed into scale_fill_gradient

Details

The output will give gray columns if there are missing values in the supplied continuous variable

Value

Returns a ggplot that you can add geoms to if you'd like

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)
otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

set %>% cor_heatmap(table = "Class", gestational_age, weight)
```

---

cor_rocky_mtn *Create Rocky Mountain plots from taxa relative abundance correlations*

Description

Calculate the correlation between the relative abundance of each taxa within a specified table and a continuous variable of interest. Correlation is calculated by cor. By default, Kendall's correlation is used to account for the prevalence of ties that often occur (lots of 0s)

Usage

```
cor_rocky_mtn(
  micro_set,
  table,
  x,
  y = clr,
  method = "spearman",
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  subtitle = NULL,
  cut_lines = TRUE,
  line_text = TRUE,
  sig_text = TRUE,
  lwd = 1,
  cor_label = 0.5,
  breaks = c(-0.6, -0.5, -0.3, 0.3, 0.5, 0.6)
)
```

Arguments

`micro_set` A tidy_micro data set

`table` OTU table of interest

`x` Continuous variable of interest

`y` The taxa information. The centered log ratio (clr) is recommended.

`method` Correlation type; must be supported by `cor`. By default it is "spearman" to use with clr. If you'd like to use taxa ra, it is recommend you switch to Kendall's correlation to account for the large number of ties common in taxa ra (lots of 0s)

`main` Plot title

`xlab` Lable for x-axis

`ylab` Label for y-axis

`subtitle` Plot subtitle

`cut_lines` Add lines for p-value cutoffs

`line_text` Label p-value cut-offs

`sig_text` Label taxa with correlations greater than `cor_label` in magnitude

`lwd` line width for cut_lines

`cor_label` Cutoff for correlations to be labeled

`breaks` Where to place cut_lines along y-axis

Value

A ggplot you can add geoms to if you'd like

Author(s)

Charlie Carpenter, Dan Frank

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

set %>% cor_rocky_mtn(table = "Family", weight, cor_label = 0.3)
```

fam *An OTU table of family level taxa counts*

Description

Infants who required mechanical ventilation had tracheal aspirates samples collected at 7, 14, and 21 days of age (+/- 48 hours). Infants who were mechanically ventilated and had at least one tracheal aspirate collected were included in this study. Subjects were required to be enrolled within 7 days of age. Bacterial profiles were determined by broad-range amplification and sequence analysis of 16S rRNA genes.

Usage

```
fam
```

Format

A 116x75 data.frame

`OTU_Name`	A character vector of family level OTU names
`Lib names`	The following columns are the sequencing counts for each library with library names

Source

https://doi.org/10.1371/journal.pone.0170120

micro_alpha_reg *Linear regression on alpha diversities within a micro_set*

Description

A simple wrapper to run standard linear regression though the `lm` function. Will only use alpha diversities distinct libraries (Lib) from the specified table as to not inflate the sample size

Usage

```
micro_alpha_reg(alpha_set, table, ...)
```

Arguments

`alpha_set`	A tidy_micro data set with alpha diversities calculated by `alpha_div`
`table`	OTU table of interest
`...`	Covariates of interest. Can include interaction terms such as `Group*Age`

Value

A data frame containing the model estimates for each alpha diversity

Note

Be aware of your minimal sequencing depth as this will be the size of all bootstrapped resamples (rarefied).

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs <- list(Phylum = phy, Class = cla, Order = ord, Family = fam)

set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including first week


set_fam_alpha <- set %>% alpha_div(table = "Family", min_depth = 5000, min_goods = 90)
set_fam_alpha %>% micro_alpha_reg(table = "Family", bpd1)
```

micro_chisq *Run Chi-Squared tests for each taxa*

Description

Run Chi-Squared tests for presence / absence of each taxa in you data set, or each taxa that didn’t converge in negative binomial models

Usage

```
micro_chisq(micro_set, table, grp_var, y = bin, mod = NULL, ...)
```

Arguments

`micro_set`	A tidy_micro data set
`table`	The OTU table you’d like to test
`grp_var`	Grouping variable for chi-squared test
`y`	Response variable for chi-squared test. Default is presence / absence (bin)
`mod`	The output from mods if you’d like to only run on taxa that did not converge
`...`	Options to be passed to chisq.test

Details

If the taxa are present or absent in every subject the chi-sqared test will not but run. The returned chi-sqared stat will either be "All Absent" or "All Present." This will be clear in the output

Value

A data from containing the taxa, the chi-squared statistic, and the p-value of the test.

References

help(chisq.test)

Examples

```
data(cla); data(clin)

set <- tidy_micro(otu_tabs = cla, tab_names = "Class", clinical = clin,
prev_cutoff = 5, ra_cutoff = 0.1, exclude_taxa = c("Unclassified", "Bacteria")) %>%
filter(day == 7) ## Only including the first week

## Chi-squared test on every taxa's presence/absence
set %>% micro_chisq(table = "Class", grp_var = bpd1,
simulate.p.value = TRUE)

## Chi-squared test on every taxa whose model didn't converge
nb_cla <- set %>% nb_mods(table = "Class", bpd1)

micro_chisq(micro_set = set, table = "Class", grp_var = bpd1,
mod = nb_cla, simulate.p.value = TRUE)
```

micro_forest *Create forest plots from negative binomial taxa models*

Description

Create forest plots for specified coefficients in negative binomial taxa models. Plots estimated beta coefficients and confidence intervals

Usage

```
micro_forest(
  modsum,
  ...,
  main,
  ylab,
  xlab,
  subtitle,
  legend_title,
  legend_labs
)
```

Arguments

`modsum`	The output from nb_mods
`...`	The covariate you'd like to plot. Must be in the models created by nb_mods
`main`	The title for your plot
`ylab`	The label for the y-axis; default is "Taxa"
`xlab`	The label for the x-axis; default is output from function "TeX"
`subtitle`	The plot subtitle
`legend_title`	The title of the plot's legend
`legend_labs`	The names of the elements within the legend

Value

Returns a ggplot that you can add geoms to if you'd like

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Creating negative binomial models on filtered tidi_micro set
nb_fam <- set %>%
otu_filter(prev_cutoff = 5, ra_cutoff = 0.1,
exclude_taxa = c("Unclassified", "Bacteria")) %>%
nb_mods(table = "Family", bpd1)

nb_fam %>% micro_forest(bpd1)
```

micro_heatmap *Create heatmaps of estiamted coefficients from negative binomial models*

Description

A function to create heatmaps of estimated beta coeffients from each model fit by nb_mods

Usage

```
micro_heatmap(
  modsum,
  low_grad,
  high_grad,
  mid_grad,
  midpoint = 0,
  top_taxa = 10,
  low_lim,
  high_lim,
  mute_cols = T,
  alpha = 0.05,
  dot_size = 2,
  dot_shape = 8,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  subtitle = NULL,
  xaxis = NULL,
  legend_title = NULL,
  caption = NULL
)
```

Arguments

`modsum`	The output from nb_mods
`low_grad`	The low gradient colors for the coefficient magnitude. Will be fed into scale_fill_gradient
`high_grad`	The high gradient colors for the coefficient magnitude. Will be fed into scale_fill_gradient
`mid_grad`	The medium gradient colors for the coefficient magnitude. Will be fed into scale_fill_gradient
`midpoint`	Midpoint for coefficient magnitude in legend
`top_taxa`	Only plot X taxa with the largest magnitude beta coefficients
`low_lim`	Lower limits of the fill gradient. Will default to the largest magnitude effect size
`high_lim`	Upper limits of the fill gradient. Will default to the largest magnitude effect size
`mute_cols`	Mute the colors of the fill gradients
`alpha`	Mark beta coefficient cells with p-values below this cutoff
`dot_size`	size of marker in cells
`dot_shape`	shape of marker in cells
`main`	Plot title
`xlab`	x-axis label
`ylab`	y-axis label
`subtitle`	Plot label
`xaxis`	Labels for the x-axis ticks
`legend_title`	Title of figure legend
`caption`	plot caption to be displayed at the bottom of plot

Details

The output will give gray columns if there are missing values in the supplied continuous variable

Value

Returns a ggplot that you can add geoms to if you'd like

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Creating negative binomial models on filtered tidy_micro set
nb_fam <- set %>%
mutate(bpd1 = factor(bpd1)) %>% ## making bpd1 a factor
otu_filter(ra_cutoff = 0.1, exclude_taxa = c("Unclassified", "Bacteria")) %>%
nb_mods(table = "Family", bpd1)

nb_fam %>% micro_heatmap
```

---

micro_pca *Calculate and plot principle components*

Description

Principle components are calculated on the centerted log ratio tranformation of the OTU table using the prcomp function from the stats package. Scaling the OTU table to a unit variance is the default option, and recommended, but this can be changed using scaled = F.

Usage

```
micro_pca(
  micro_set,
  table = NULL,
  dist = NULL,
  grp_var,
  y = clr,
  scale = TRUE,
  axes_arrows = F,
  main = NULL,
  subtitle = NULL,
  legend_title = NULL
)
```

Arguments

micro_set	A tidy_micro data set
table	OTU table of interest

`dist`	A distance matrix, such as a beta diversity. If supplied a PCoA plot will be returned
`grp_var`	Categorical grouping variable for color
`y`	Value to calculate principle components or coordinates on. Default is centered log ratio (recommended)
`scale`	Logical. Indicating whether the variables should be scaled to have unit variance before the analysis takes place
`axes_arrows`	Logical. Plot component axes arrows
`main`	Plot title
`subtitle`	Plot subtitle
`legend_title`	Legend title

Details

PCA calculation is done by a singular value decomposition of the (centered and possibly scaled) data matrix, not by using eigen on the covariance matrix. This is generally the preferred method for numerical accuracy. Calculations are accomplished through the `prcomp` function, and the plot is created through internal code based on the ggbiplot function `https://github.com/vqv/ggbiplot`.

Value

A ggplot you can add geoms to if you'd like

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

Mardia, K. V., J. T. Kent, and J. M. Bibby (1979) Multivariate Analysis, London: Academic Press.

Venables, W. N. and B. D. Ripley (2002) Modern Applied Statistics with S, Springer-Verlag.

Vincent Q. Vu (2011). ggbiplot: A ggplot2 based biplot. `https://github.com/vqv/ggbiplot`

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs <- list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including first week

## PCA Plot
set %>% micro_pca(table = "Family", grp_var = bpd1)

## PCoA Plot (Recommended for p > n)

bray_beta <- set %>% beta_div(table = "Family")

set %>%
micro_pca(dist = bray_beta, grp_var = bpd1)
```

micro_PERMANOVA *A function to run PERMANOVA on tidi_micro data sets*

Description

A wrapper function to call adonis2 from the vegan package. PERMANOVA is a method for partitioning distance matrices among sources of variation and fitting linear models (e.g., factors, polynomial regression) to distance matrices; uses a permutation test with pseudo-F ratios

Usage

```
micro_PERMANOVA(micro_set, beta_div, method, ..., nperm = 999)
```

Arguments

micro_set	A tidy_micro data set
beta_div	A dissimilarity matrix calculated by beta_div
method	A character string indicating the method used to calculated dissimilarity
...	Covariates of interest
nperm	Number of permutations

Details

The function adonis2 is based on the principles of McArdle & Anderson (2001) and can perform sequential, marginal and overall tests. Function adonis2 also allows using additive constants or squareroot of dissimilarities to avoid negative eigenvalues

References

vegdist adonis2

See Also

adonis

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)
otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Bray-Curtis beta diversity
bray <- set %>% beta_div(table = "Family")

set %>% micro_PERMANOVA(bray, method = "bray", bpd1)
```

micro_rank_sum *Run rank sum tests for each taxa within an OTU table*

Description

Runs a rank sum test for each taxa within an OTU table or each taxa that didn't converge in `nb_mods` or `bb_mods`

Usage

```
micro_rank_sum(micro_set, table, grp_var, y = ra, mod = NULL, ...)
```

Arguments

`micro_set`	A tidy_micro data set
`table`	OTU table of interest
`grp_var`	A factor variable for grouping
`y`	A continuous response variable. Taxa relative abundance (ra) is recommended
`mod`	The output from `nb_mods` or `bb_mods` if desired
`...`	Options to be passed to `wilcox.test` or `kruskal.test`

Details

The grp_var must have a least 2 levels. For a 2 level factor a Mann-Whitney test will be calculated through `wilcox.test`, and for 3 or more levels a Kruskal-Wallis test will be run throuh `kruskal.test`

Value

A data frame containing the p-value for each taxa's rank sum test.

References

`kruskal.test` and `wilcox.test`

Examples

```
data(cla); data(clin)

set <- tidy_micro(otu_tabs = cla, tab_names = "Class", clinical = clin,
prev_cutoff = 5, ra_cutoff = 0.1, exclude_taxa = c("Unclassified", "Bacteria")) %>%
filter(day == 7) ## Only including the first week

## Rank sum test on every taxa's relative abundance
set %>% micro_rank_sum(table = "Class", grp_var = bpd1)

## Rank sum test on every taxa whose model didn't converge
```

```
nb_cla <- nb_mods(set, table = "Class", bpd1)

micro_rank_sum(micro_set = set, table = "Class",
grp_var = bpd1, mod = nb_cla)
```

micro_rocky_mtn *Create Rocky Mountain plots from negative binomial taxa models*

Description

Display the magnitude of log p-values for each of the taxa in nb_mods as vertical bars next to each other along the x-axis. The direction of the bars will be determined by the direction of the estimated relationship. The taxa will be color coded by the phylum they belong to, and taxa that have FRD adjusted p-values below your desired significance cutoff for the specified covariate will be labeled

Usage

```
micro_rocky_mtn(
  modsum,
  ...,
  main = NULL,
  ylab = NULL,
  subtitle = NULL,
  pval_lines = TRUE,
  pval_text = TRUE,
  sig_text = TRUE,
  facet_labels = NULL,
  alpha = 0.05,
  lwd = 2,
  lty = 1
)
```

Arguments

modsum	The output from nb_mods
...	The covariate you'd like to plot. Must be in the models created by nb_mods
main	Plot title
ylab	y-axis labels
subtitle	Plot subtitle
pval_lines	Logical; include horizonal dashed lines at corresponding p-values
pval_text	Logical; label the y-axis with corresponding p-values
sig_text	Logical; label the taxa with p-values below specified alpha
facet_labels	Labels for different facets if covariate has more than 1 beta coefficient
alpha	Significance cutoff
lwd	Line width for pval_lines
lty	Line type for pval_lines

Value

A ggplot you can add geoms to if you'd like

Author(s)

Charlie Carpenter, Rachel Johnson, Dan Frank

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Creating negative binomial models on filtered tidy_micro set
nb_fam <- set %>%
otu_filter(ra_cutoff = 0.1, exclude_taxa = c("Unclassified", "Bacteria")) %>%
nb_mods(table = "Family", bpd1)

nb_fam %>% micro_rocky_mtn(bpd1)
```

nb_bars *Create stacked bar charts based on negative binomial model estimates*

Description

nb_bars takes the output from nb_mods and creates stacked bar charts of the estimated relative abundance for each taxa. The benefit of modeling each taxa before created stacked bar charts is the ability to control for potential confounders. The function will facet wrap interaction terms. Currently, only quant_style "discrete" can be used for an interaction between two quantitative variables

Usage

```
nb_bars(
  modsum,
  ...,
  range,
  quant_style = c("continuous", "discrete"),
  top_taxa = 0,
  RA = 0,
  specific_taxa = NULL,
  lines = TRUE,
  xaxis,
  main,
  subtitle,
  xlab,
  ylab,
  facet_labels = NULL,
  facet_layout = 1
)
```

Arguments

`modsum` The output from nb_mods

`...` The covariate you'd like to plot. Can be an interaction term or main effect, but must be in the models created by nb_mods

`range` The range you'd like to plot over for a quantitative variable. Will default to the IQR

`quant_style` "continuous" will plot over the entire range specified; "discrete" will plot only the endpoints of the range specified. "continuous" by default. This option is ignored without a quantitative variable

`top_taxa` Only plot X taxa with the highest relative abundance. The rest will be aggregated into an "Other" category

`RA` Only plot taxa with a relative abundance higher than X. The rest will be aggregated into an "Other" category

`specific_taxa` Plot this specific taxa even if it doesn't meet the top_taxa or RA requirements

`lines` Logical; Add outlines around the different taxa colors in the stacked bar charts

`xaxis` Labels for the x-axis ticks. Most useful for categorical variables and defaults to the levels

`main` Plot title

`subtitle` Subtitle for the plot

`xlab` x-axis label

`ylab` y-axis label

`facet_labels` Labels for the facets created for interaction terms

`facet_layout` Rearrange the facets created for interaction terms

Value

Returns a ggplot that you can add geoms to if you'd like

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)
otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Creating negative binomial models on filtered tidy_micro set
nb_fam <- set %>%
otu_filter(ra_cutoff = 0.1, exclude_taxa = c("Unclassified", "Bacteria")) %>%
nb_mods(table = "Family", bpd1)

nb_fam %>%
nb_bars(bpd1, top_taxa = 9, xlab = "BPD Severity")
```

nb_mods *Fit negative binomial models to each taxa within an OTU table*

Description

Fit negative binomial models to each taxa within an OTU table through glm.nb in the **MASS** package. Models can include a random effect if desired. Modesl will then be fit through glmer.nb in the lmer package. Summaries for models or confidence intervals that fail to converge will not be returned, but taxa summaries will be provided in the output. Rank-Sum tests or presence/absence tests can be run on these taxa using tidi_rank_sum or tidi_chisq, respectively

Usage

```
nb_mods(
  micro_set,
  table,
  ...,
  Offset = TRUE,
  ref = NULL,
  SS_type = c(2, 3, "II", "III")
)
```

Arguments

micro_set	A tidy_micro data set
table	OTU table of interest
...	Covariates of interest. Can be interactions such as Group*Age
Offset	Logical; include subject sequencing depth as an offset for negative binomial models. This is highly recommended
ref	A character vector of the desired reference levels for each factor covariate. The order of the specifed references must match the order for the corresponding covariates specified in '...'
SS_type	Type of sums of squares calculated in Anova. Either type II (2) or type III (3) sums of squares

Details

Models containing only fixed effects are fit using glm.nb in the **MASS** package and models containing random effects are fit using glmer.nb. ANOVA / ANCOVA tests are conducted using a Likelihood Ratio test for fixed effects models and Chi-Squared tests for random effect models.

Value

A list containing several different model components and summaries

`Convergend_Summary`
A data.frame of model summaries from convergent models. Includes the Taxa name, the model coefficient, the estimated beta, the beta's 95 percent confidence interval, Z score, p_value, false discovery rate p-value, and p-value from likelihood ratio test

`Estimate_Summary`
A data.frame of model estimates from convergent models intended to be ready for export for publications. Includes the Taxa name, the model coefficient, the estimated Rate Ratio, the Wald 95 percent confidence interval, the Z-score, and false discovery rate p-value

`RA_Summary` A data.frame of taxa summaries. Includes the Taxa name, grouping variables (each factor variable in your models), sample size (n), percent of 0 counts, basic summaries of relative abundance, percentiles of relative abundance, and a logical indicator of whether or not the model converged

`formula` The formula used in the model

`Model_Coef` Model coefficients (used in plotting funcitons)

`Model_Covs` Model covariates (used in plotting functions)

Note

False Discovery Rate p-values are calculated using `p.adjust`. Estimated rate ratios and confidence intervals for interactions in the Estimate_Summary table include all main effects. It is not simply the exponentiated interaction beta, it is the interaction of the sum of the intercept, corresponding main effect betas, and interaction betas

References

`Anova`, `glm.nb`, `glmer.nb`

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs <- list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7)

nb_fam <- set %>%
otu_filter(prev_cutoff = 5, ra_cutoff = 0.1, exclude_taxa = c("Unclassified", "Bacteria")) %>%
nb_mods(table = "Family", bpd1)

names(nb_fam)
nb_fam$Estimate_Summary
```

ord *An OTU table of order level taxa counts*

Description

Infants who required mechanical ventilation had tracheal aspirates samples collected at 7, 14, and 21 days of age (+/- 48 hours). Infants who were mechanically ventilated and had at least one tracheal aspirate collected were included in this study. Subjects were required to be enrolled within 7 days of age. Bacterial profiles were determined by broad-range amplification and sequence analysis of 16S rRNA genes.

Usage

```
ord
```

Format

A 62x75 data.frame

`OTU_Name` A character vector of ord level OTU names

`Lib names` The following columns are the sequencing counts for each library with library names

Source

https://doi.org/10.1371/journal.pone.0170120

otu_filter *A function to aggregate low prevalence, abundance, or unwanted taxa together*

Description

Will take a tidi_micro set and aggregate the raw counts of taxa with a low prevalence and/or abundance into a new "Other" taxa. Can also find specific taxa you'd like to include in the "Other" taxa counts. Once the counts are aggregated taxa relative abundance, centered log ratio (CLR) transformations, and presence will be recalculated. This recalculation will only change the "Other" category

Usage

```
otu_filter(
  micro_set,
  prev_cutoff = 0,
  ra_cutoff = 0,
  exclude_taxa = NULL,
  filter_summary = T
)
```

Arguments

`micro_set`	A tidy_micro data set
`prev_cutoff`	Minimum percent of subjects with OTU counts above 0
`ra_cutoff`	At leat one subject must have RA above this subject
`exclude_taxa`	A character vector of OTU names that you would like filter into your "Other" category
`filter_summary`	Logical; print out summaries of filtering steps

Details

$\frac{1}{Total}$ will be added to each taxa count for CLR tranformations in order to avoid issues with log(0)

Value

Returns a tidy_micro set

Author(s)

Charlie Carpenter and Dan Frank

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

filter_set <- set %>%
otu_filter(prev_cutoff = 5, ## 5% of subjects must have this bug, or it is filtered
  ra_cutoff = 1, ## At least 1 subject must have RA of 1, or it is filtered
  exclude_taxa = c("Unclassified", "Bacteria") ## Unclassified taxa we don't want
)
```

pca_3d *Create 3d PCA plots*

Description

Create three dimensional PCA plots from longitudinal data or multiple omics data sets.

Usage

```
pca_3d(
  micro_set,
  table,
  time_var,
  subject,
  y = clr,
  modes = c("AC", "BA", "CB"),
  dist_method = "euclidean",
  type = "PCoA",
  plot_scores = FALSE,
  n_compA,
  n_compB,
  n_compC,
  cex.axis = 1,
  cex.lab = 1,
  main = NULL,
  subtitle = NULL,
  scalewt = TRUE
)
```

Arguments

micro_set	A tidy_micro data set
table	OTU table of interest
time_var	The time point variable column name in your tidi_MIBI set
subject	The subject variable column name in your tidi_MIBI set
y	Value to calculate principle components or coordinates on. Default is centered log ratio (recommended)
modes	Components of the data to focus on: time, subjects, bacteria, etc. "AC" by default
dist_method	Dissimilartiy method to be calculated by vegdist. Euclidean by default
type	"PCA" for principle components or "PCoA" to calculated dissimilarity matrix using vegdist
plot_scores	Plot the scores instead of the principle components
n_compA	The number of components along first axis. See details
n_compB	The number of components along second axis. See details
n_compC	The number of components along third axis. See details
cex.axis	Options for scatterplot3d
cex.lab	Options for scatterplot3d
main	Plot title
subtitle	Plot subtitle
scalewt	Logical; center and scale OTU table, recommended

Details

Requires that you have columns for subject name and time point. Data must be complete across time points. The function will filter out inconsistent subjects

When type = "PCoA" the component matrices must be specified prior to the optimization. This is handled automatically.

If n_compA, n_compB, and n_compC aren't specified they will default to the number of complete subjects, the number of taxa, and the number of time points, respectively. This slows down performance slightly, but will not change the results.

Author(s)

Charlie Carpenter, Kayla Williamson

References

`vegdist`

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin)

set %>% pca_3d(table = "Family", time_var = day, subject = study_id)
```

`phy` *An OTU table of phylum level taxa counts*

Description

Infants who required mechanical ventilation had tracheal aspirates samples collected at 7, 14, and 21 days of age (+/- 48 hours). Infants who were mechanically ventilated and had at least one tracheal aspirate collected were included in this study. Subjects were required to be enrolled within 7 days of age. Bacterial profiles were determined by broad-range amplification and sequence analysis of 16S rRNA genes.

Usage

```
phy
```

Format

A 15x75 data.frame

`OTU_Name` A character vector of phylum level OTU names

`Lib names` The following columns are the sequencing counts for each library with library names

Source

https://doi.org/10.1371/journal.pone.0170120

---

ra_bars *Function to make stacked bar charts of taxa relative abundance*

Description

A function to make stacked bar charts of taxa relative abuncance with the choice to stratify by a variable of interest

Usage

```
ra_bars(
  micro_set,
  table,
  ...,
  top_taxa = 0,
  RA = 0,
  specific_taxa,
  main,
  subtitle,
  ylab,
  xlab,
  xaxis,
  lines = TRUE
)
```

Arguments

micro_set	A tidy_micro data set
table	OTU table you'd like to use when calculating alpha diversity. Your lowest level is recommended
...	A categorical variable by which you'd like to stratify your relative abundances
top_taxa	Only plot X taxa with the highest relative abundance. The rest will be aggregated into an "Other" category.
RA	Only plot taxa with a relative abundance higher than X. The rest will be aggregated into an "Other" category.
specific_taxa	Plot this specific taxa even if it doesn't meet the top_taxa or RA requirements
main	Plot title
subtitle	Subtitle for the plot
ylab	y-axis label
xlab	x-axis label
xaxis	Labels for the x-axis ticks. Most useful for categorical variables and defaults to the levels
lines	Logical; Add outlines around the different taxa colors in the stacked bar charts

Value

Returns a ggplot that you can add geoms to if you'd like

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

## Full cohort abundance
set %>%
ra_bars(table = "Family", top_taxa = 10)

## Stratified by variable of interest
set %>%
ra_bars(table = "Family", bpd1, top_taxa = 10)
```

`taxa_boxplot` *Function to make boxplots of taxa counts or relative abundance*

Description

A function to make boxplots of one specified taxa relative abundance with the option to stratify by a factor variable

Usage

```
taxa_boxplot(
  micro_set,
  taxa,
  ...,
  y = ra,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  subtitle = NULL,
  legend_title = NULL
)
```

Arguments

`micro_set`	A tidy_micro data set
`taxa`	A character string. The name of the taxa of interest
`...`	The factor variable you'd like to stratify by
`y`	The taxa information

xlab	x-axis label
ylab	y-axis label
main	Plot title
subtitle	Subtitle for the plot
legend_title	Title of plot legend

Value

A ggplot that you can add geoms to if you'd like

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)
otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
filter(day == 7) ## Only including the first week

set %>%
taxa_boxplot("Firmicutes/Bacilli/Bacillales/Staphylococcaceae", bpd1)
```

---

taxa_summary *Summarize the information*

Description

Give taxa summary table stratified by variables of interest and/or OTU tables

Usage

```
taxa_summary(micro_set, ..., table = NULL, obj = ra, taxa = TRUE)
```

Arguments

micro_set	A tidy_micro data set
...	Covariates of interest
table	OTU table of interest. If NULL, all tables will be used
obj	The taxonomic information of interest
taxa	Logical; Whether or not to stratify by taxa

Value

A tibble containing columns of stratifying variables and several summary columns

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs <- list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin) %>%
mutate(bpd1 = factor(bpd1))

## Summarize each taxa by Table
set %>% taxa_summary

## Summarize each taxa by a categorical variable of interest
set %>% taxa_summary(bpd1)

## Summarize each taxa by a categorical variable of interest within a Table
set %>% taxa_summary(bpd1, table = "Phylum")

## Summarize within group or table only
set %>% taxa_summary(taxa = FALSE)
```

`three_mode` *Create Three Mode PCA and PCoA plots*

Description

Three Mode Principal Components, an ordination method that can take into account repeated measure of subjects. These methods have also been extended to other common ecological distance metrics for Three Mode Principal Coordinate Analysis

Usage

```
three_mode(
  micro_set,
  table,
  group,
  time_var,
  subject,
  y = clr,
  modes = c("AC", "BA", "CB"),
  plot_scores = F,
  n_compA,
  n_compB,
  n_compC,
  main = NULL,
  subtitle = NULL,
  legend_title = NULL,
  scalewt = TRUE
)
```

Arguments

`micro_set`	A tidy_micro data set
`table`	OTU table of interest
`group`	A categorical variable to color by
`time_var`	The time point variable column name in your tidi_MIBI set
`subject`	The subject variable column name in your tidi_MIBI set
`y`	Value to calculate principle components or coordinates on. Default is centered log ratio (recommended)
`modes`	Components of the data to focus on: time, subjects, bacteria, etc.
`plot_scores`	Plot the scores instead of the principle components
`n_compA`	The number of components along first axis. See details
`n_compB`	The number of components along second axis. See details
`n_compC`	The number of components along third axis. See details
`main`	Plot title
`subtitle`	Plot subtitle
`legend_title`	Plot legend title
`scalewt`	Logical; center and scale OTU table, recommended

Details

Requires that you have columns for subject name and time point. Data must be complete across time points. The function will filter out inconsistent subjects

If n_compA, n_compB, and n_compC aren't specified they will default to the number of complete subjects, the number of taxa, and the number of time points, respectively. This slows down performance slightly, but will not change the results.

Value

A ggplot you can add geoms to if you'd like

Author(s)

Charlie Carpenter, Kayla Williamson

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

otu_tabs = list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin)

set %>% three_mode(table = "Family", group = bpd1, time_var = day, subject = study_id)
```

`tidy_micro` *A function to merge multiple OTU tables and clinical data into a "tidy" format*

Description

A function to take any number of OTU tables (or other sequencing data tables), calculate taxa prevalence, relative abundance, and a CLR transformation, and finally merges clinical data

Usage

```
tidy_micro(
  otu_tabs,
  tab_names,
  clinical,
  prev_cutoff = 0,
  ra_cutoff = 0,
  exclude_taxa = NULL,
  library_name = "Lib",
  complete_clinical = T,
  filter_summary = T
)
```

Arguments

`otu_tabs` A single table or list of metagenomic sequencing data. Tables should have a first column of OTU Names and following columns of OTU counts. Column names should be sequencing library names

`tab_names` names for otu_tabs. These will become the "Tables" column. It is also an option to simply name the OTU tables in the list supplied to otu_tabs

`clinical` Sequencing level clinical data. Must have a column with unique names for library (sequencing ID)

`prev_cutoff` A prevalence cutoff where *X* percent of libraries must have this taxa or it will be included in the "Other" category

`ra_cutoff` A relative abundance (RA) cutoff where at least one library must have a RA above the cutoff or the taxa will be included in the "Other" category

`exclude_taxa` A character vector used to specify any taxa that you would like to included in the "Other" category. Taxa specified will be included in "Other" for every OTU table provided

`library_name` The column name containing sequencing library names. Should match with column names of supplied OTU tables (after first column)

`complete_clinical` Logical; only include columns from OTU tables who's library name is in clinical data

`filter_summary` Logical; print out summaries of filtering steps

Details

Column names of the OTU tables must be the same for each table, and these should be the the library names inside of your clinical. Please see the vignette for a detailed description.

The CLR transformation adds (1 / sequencing depth) to each OTU count for each library before centering and log transforming in order to avoid issues with 0 counts.

The list of OTU tables are split, manipulated, and stacked into a data frame using the `ldply` function from the **plyr** package. Names of OTU tables supplied will be the name of their "Table" in the final tidy_micro set

Value

A data.frame in the tidy_micro format

Author(s)

Charlie Carpenter

Examples

```
data(phy); data(cla); data(ord); data(fam); data(clin)

## Multiple OTU tables with named list
otu_tabs <- list(Phylum = phy, Class = cla, Order = ord, Family = fam)
set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin)

## Multiple OTU tables with unnamed list
unnamed_tabs <- list(phy,cla,ord,fam)
set <- tidy_micro(otu_tabs = unnamed_tabs,
tab_names = c("Phylum", "Class", "Order", "Family"), clinical = clin)

## Single OTU table
set <- tidy_micro(otu_tabs = cla, tab_names = "Class", clinical = clin)

## Filtering out low abundance or uninteresting taxa right away
## WARNING: Only do this if you do not want to calculate alpha diversities with this micro_set

filter_set <- tidy_micro(otu_tabs = otu_tabs, clinical = clin,
              prev_cutoff = 5, ## 5% of libraries must have this bug, or it is filtered
             ra_cutoff = 1, ## At least 1 libraries must have RA of 1, or it is filtered
          exclude_taxa = c("Unclassified", "Bacteria") ## Unclassified taxa we don't want
              )
```

Index