Package ‘microbiome’

January 14, 2024

Type Package
Title Microbiome Analytics
Description Utilities for microbiome analysis.
Encoding UTF-8
Version 1.24.0
Date 2022-04-22
biocViews Metagenomics,Microbiome,Sequencing,SystemsBiology
License BSD_2_clause + file LICENSE
Depends R (>= 3.6.0), phyloseq, ggplot2
Imports Biostrings, compositions, dplyr, reshape2, Rtsne, scales, stats, tibble, tidyr, utils, vegan
Suggests BiocGenerics, BiocStyle, Cairo, knitr, rmarkdown, testthat
URL http://microbiome.github.io/microbiome
MailingList microbiome <microbiome-devel@googlegroups.com>
BugReports https://github.com/microbiome/microbiome/issues
VignetteBuilder knitr
RoxygenNote 7.2.1
git_url https://git.bioconductor.org/packages/microbiome
git_branch RELEASE_3_18
git_last_commit 4e08c43
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-01-14
Author Leo Lahti [aut, cre] (<https://orcid.org/0000-0001-5537-637X>), Sudarshan Shetty [aut] (<https://orcid.org/0000-0001-7280-9915>)
Maintainer Leo Lahti <leo.lahti@iki.fi>
R topics documented:

- microbiome-package ........................................... 4
- abundances ......................................................... 4
- add_besthit ......................................................... 5
- add_refseq ......................................................... 6
- aggregate_rare .................................................... 7
- aggregate_taxa .................................................... 8
- alpha ............................................................... 9
- associate .......................................................... 10
- atlas1006 .......................................................... 11
- baseline ............................................................ 12
- bimodality ........................................................ 13
- bimodality_sarle .................................................. 15
- boxplot_abundance .............................................. 16
- boxplot_alpha ..................................................... 17
- chunk_reorder .................................................... 18
- cmat2table ......................................................... 19
- collapse_replicates ............................................ 20
- core ............................................................... 21
- core_abundance .................................................. 22
- core_heatmap ..................................................... 23
- core_matrix ....................................................... 24
- core_members ..................................................... 25
- coverage .......................................................... 26
- default_colors .................................................. 27
- densityplot ....................................................... 27
- dietswap .......................................................... 29
- divergence ......................................................... 30
- diversity .......................................................... 31
- dominance ......................................................... 32
- dominant .......................................................... 34
- estimate_stability .............................................. 35
- evenness .......................................................... 36
- find_optima ....................................................... 37
- gktau ............................................................... 38
- group_age ........................................................ 39
- group_bmi ........................................................ 39
- heat ................................................................. 41
- hitchip.taxonomy ............................................... 44
- hotplot ............................................................. 45
- inequality ........................................................ 46
- intermediate_stability ......................................... 47
- is_compositional ............................................... 48
- log_modulo_skewness ........................................... 49
- low_abundance ................................................... 50
- map_levels ......................................................... 51
- merge_taxa2 ...................................................... 52
<table>
<thead>
<tr>
<th>R topics documented:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>meta</td>
<td>53</td>
</tr>
<tr>
<td>multimodality</td>
<td>53</td>
</tr>
<tr>
<td>neat</td>
<td>55</td>
</tr>
<tr>
<td>neatsort</td>
<td>56</td>
</tr>
<tr>
<td>overlap</td>
<td>57</td>
</tr>
<tr>
<td>peerj32</td>
<td>58</td>
</tr>
<tr>
<td>plot_atlas</td>
<td>59</td>
</tr>
<tr>
<td>plot_composition</td>
<td>60</td>
</tr>
<tr>
<td>plot_core</td>
<td>61</td>
</tr>
<tr>
<td>plot_density</td>
<td>62</td>
</tr>
<tr>
<td>plot_frequencies</td>
<td>63</td>
</tr>
<tr>
<td>plot_landscape</td>
<td>64</td>
</tr>
<tr>
<td>plot_regression</td>
<td>66</td>
</tr>
<tr>
<td>plot_taxa_prevalence</td>
<td>68</td>
</tr>
<tr>
<td>plot_tipping</td>
<td>69</td>
</tr>
<tr>
<td>potential_analysis</td>
<td>70</td>
</tr>
<tr>
<td>potential_univariate</td>
<td>71</td>
</tr>
<tr>
<td>prevalence</td>
<td>73</td>
</tr>
<tr>
<td>psmelt2</td>
<td>74</td>
</tr>
<tr>
<td>quiet</td>
<td>75</td>
</tr>
<tr>
<td>radial_theta</td>
<td>75</td>
</tr>
<tr>
<td>rare</td>
<td>76</td>
</tr>
<tr>
<td>rare_abundance</td>
<td>77</td>
</tr>
<tr>
<td>rare_members</td>
<td>78</td>
</tr>
<tr>
<td>rarity</td>
<td>79</td>
</tr>
<tr>
<td>readcount</td>
<td>80</td>
</tr>
<tr>
<td>read_biom2phyloseq</td>
<td>81</td>
</tr>
<tr>
<td>read_csv2phyloseq</td>
<td>82</td>
</tr>
<tr>
<td>read_mothur2phyloseq</td>
<td>83</td>
</tr>
<tr>
<td>read_phyloseq</td>
<td>84</td>
</tr>
<tr>
<td>remove_samples</td>
<td>85</td>
</tr>
<tr>
<td>remove_taxa</td>
<td>86</td>
</tr>
<tr>
<td>richness</td>
<td>87</td>
</tr>
<tr>
<td>spreadplot</td>
<td>88</td>
</tr>
<tr>
<td>summarize_phyloseq</td>
<td>89</td>
</tr>
<tr>
<td>taxa</td>
<td>89</td>
</tr>
<tr>
<td>TibbleUtilities</td>
<td>90</td>
</tr>
<tr>
<td>timesplit</td>
<td>91</td>
</tr>
<tr>
<td>time_normalize</td>
<td>92</td>
</tr>
<tr>
<td>time_sort</td>
<td>92</td>
</tr>
<tr>
<td>top</td>
<td>93</td>
</tr>
<tr>
<td>top_taxa</td>
<td>94</td>
</tr>
<tr>
<td>transform</td>
<td>95</td>
</tr>
<tr>
<td>ztransform</td>
<td>96</td>
</tr>
</tbody>
</table>

Index 98
Description

Brief summary of the microbiome package

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>microbiome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>See sessionInfo() or DESCRIPTION file</td>
</tr>
<tr>
<td>Date:</td>
<td>2014-2017</td>
</tr>
<tr>
<td>License:</td>
<td>FreeBSD</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>

R package for microbiome studies

Author(s)

Leo Lahti et al. <microbiome-admin@googlegroups.com>

References

See citation('microbiome') [http://microbiome.github.io](http://microbiome.github.io)

Examples

```
citation('microbiome')
```

Description

Retrieves the taxon abundance table from phyloseq-class object and ensures it is systematically returned as taxa x samples matrix.

Usage

```
abundances(x, transform = "identity")
```
**add_besthit**

Arguments

x phyloseq-class object

transform Transformation to apply. The options include: 'compositional' (ie relative abundance), 'Z', 'log10', 'log10p', 'hellinger', 'identity', 'clr', 'alr', or any method from the vegan::decostand function.

Value

Abundance matrix (OTU x samples).

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')

Examples

data(dietswap)
a <- abundances(dietswap)
# b <- abundances(dietswap, transform='compositional')

---

**Description**

Add the lowest classification for an OTU or ASV.

**Usage**

add_besthit(x, sep = ":")

**Arguments**

x phyloseq-class object

sep separator e.g. ASV161:Roseburia

**Details**

Most commonly it is observed that taxa names are either OTU ids or ASV ids. In such cases it is useful to know the taxonomic identity. For this purpose, best_hist identifies the best available taxonomic identity and adds it to the OTU ids or ASV ids. If genus and species columns are present in input the function internally combines the names.
Description

Utility to add refseq slot for dada2 based phyloseq Object. Here, the taxa_names which are unique sequences, are stored in refseq slot of phyloseq. Sequence ids are converted to ids using tag option.

Usage

```r
add_refseq(x, tag = "ASV")
```

Arguments

- `x` : `phyloseq-class` object with sequences as rownames.
- `tag` : Provide name for Ids, Default="ASV".

Value

`phyloseq-class` object

Author(s)

Contact: Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

Examples

```r
# ps <- add_refseq(p0, tag="ASV")
# ps
```
aggregate_rare

Aggregate Rare Groups

Description
Combining rare taxa.

Usage
aggregate_rare(x, level, detection, prevalence, include.lowest = FALSE, ...)

Arguments
x phyloseq-class object
level Summarization level (from rank_names(pseq))
detection Detection threshold for absence/presence (strictly greater by default).
prevalence Prevalence threshold (in [0, 1]). The required prevalence is strictly greater by default. To include the limit, set include.lowest to TRUE.
include.lowest Include the lower boundary of the detection and prevalence cutoffs. FALSE by default.
... Arguments to pass.

Value
phyloseq-class object

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References
See citation('microbiome')

Examples
data(dietswap)
s <- aggregate_rare(dietswap, level = 'Phylum',
    detection = 0.1/100, prevalence = 5/100)
aggregate_taxa  Aggregate Taxa

Description
Summarize phyloseq data into a higher phylogenetic level.

Usage
aggregate_taxa(x, level, verbose = FALSE)

Arguments
x  phyloseq-class object
level Summarization level (from rank_names(pseq))
verbose  verbose

Details
This provides a convenient way to aggregate phyloseq OTUs (or other taxa) when the phylogenetic
tree is missing. Calculates the sum of OTU abundances over all OTUs that map to the same higher-
level group. Removes ambiguous levels from the taxonomy table. Returns a phyloseq object with
the summarized abundances.

Value
Summarized phyloseq object

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References
See citation('microbiome')

Examples
data(dietswap)
s <- aggregate_taxa(dietswap, 'Phylum')
Global Ecosystem State Variables

Description
Global indicators of the ecosystem state, including richness, evenness, diversity, and other indicators

Usage
alpha(x, index = "all", zeroes = TRUE)

Arguments
x A species abundance vector, or matrix (taxa/features x samples) with the absolute count data (no relative abundances), or phyloseq-class object
index Default is ‘NULL’, meaning that all available indices will be included. For specific options, see details.
zeroes Include zero counts in the diversity estimation.

Details
This function returns various indices of the ecosystem state. The function is named alpha (global in some previous versions of this package) as these indices can be viewed as measures of alpha diversity. The function uses default choices for detection, prevalence and other parameters for simplicity and standardization. See the individual functions for more options. All indicators from the richness, diversity, evenness, dominance, and rarity functions are available. Some additional measures, such as Chao1 and ACE are available via estimate_richness function in the phyloseq package but not included here. The index names are given the prefix richness_, evenness_, diversity_, dominance_, or rarity_ in the output table to avoid confusion between similarly named but different indices (e.g. Simpson diversity and Simpson dominance). All parameters are set to their default. To experiment with different parameterizations, see the more specific index functions (richness, diversity, evenness, dominance, rarity).

Value
A data.frame of samples x alpha diversity indicators

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References
See citation('microbiome')

See Also
dominance, rarity, phyloseq::estimate_richness
Examples

data(dietswap)
d <- alpha(dietswap, index='shannon')
# d <- alpha(dietswap, index='all')

associate

Cross Correlation Wrapper

Description

Cross-correlate columns of the input matrices.

Usage

associate(
x, 
y = NULL, 
method = "spearman", 
p.adj.threshold = Inf, 
cth = NULL, 
order = FALSE, 
n.signif = 0, 
mode = "table", 
p.adj.method = "fdr", 
verbose = FALSE, 
filter.self.correlations = FALSE
)

Arguments

x      matrix (samples x features if annotation matrix)
y      matrix (samples x features if cross-correlated with annotations)
method association method (‘pearson’, or ‘spearman’ for continuous)
p.adj.threshold q-value threshold to include features
cth     correlation threshold to include features
order    order the results
n.signif minimum number of significant correlations for each element
mode Specify output format (‘table’ or ’matrix’)
p.adj.method p-value multiple testing correction method. One of the methods in p.adjust function (‘BH’ and others; see help(p.adjust)). Default: ‘fdr’
verbose verbose
filter.self.correlations Filter out correlations between identical items.
Details

The p-values in the output table depend on the method. For the spearman and pearson correlation values, the p-values are provided by the default method in the cor.test function.

Value

List with cor, pval, pval.adjusted

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')

Examples

data(peerj32)
d1 <- peerj32$microbes[1:20, 1:10]
d2 <- peerj32$lipids[1:20,1:10]
c <- associate(d1, d2, method='pearson')

Description

This data set contains genus-level microbiota profiling with HITChip for 1006 western adults with no reported health complications, reported in Lahti et al. (2014) https://doi.org/10.1038/ncomms5344.

Usage

data(atlas1006)

Format

The data set in phyloseq-class format.

Details

The data is also available for download from the Data Dryad http://doi.org/10.5061/dryad.pk75d.

Value

Loads the data set in R.
baseline

Author(s)
Leo Lahti <microbiome-admin@googlegroups.com>

References
Lahti et al. Tipping elements of the human intestinal ecosystem. Nature Communications 5:4344, 2014. To cite the microbiome R package, see citation('microbiome')

baseline

Pick Baseline Timepoint Samples

Description
Identify and select the baseline timepoint samples in a phyloseq object.

Usage
baseline(x, na.omit = TRUE)

Arguments
x     phyloseq object. Assuming that the sample_data(x) has the fields 'time', 'sample' and 'subject'
na.omit Logical. Ignore samples with no time point information. If this is FALSE, the first sample for each subject is selected even when there is no time information.

Details
Arranges the samples by time and picks the first sample for each subject. Compared to simple subsetting at time point zero, this checks NAs and possibility for multiple samples at the baseline, and guarantees that a single sample per subject is selected.

Value
Phyloseq object with only baseline time point samples selected.

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References
See citation('microbiome')

Examples
data(peerj32)
a <- baseline(peerj32$phyloseq)
bimodality  Bimodality Analysis

Description

Estimate bimodality scores.

Usage

bimodality(
  x,
  method = "potential_analysis",
  peak.threshold = 1,
  bw.adjust = 1,
  bs.iter = 100,
  min.density = 1,
  verbose = TRUE
)

Arguments

x  A vector, matrix, or a phyloseq object
method  bimodality quantification method ("potential_analysis", 'Sarle.finite.sample', or 'Sarle.asymptotic'). If method='all', then a data.frame with all scores is returned.
peak.threshold  Mode detection threshold
bw.adjust  Bandwidth adjustment
bs.iter  Bootstrap iterations
min.density  minimum accepted density for a maximum; as a multiple of kernel height
verbose  Verbose

Details

• Sarle.finite.sample Coefficient of bimodality for finite sample. See SAS 2012.
• Sarle.asymptotic Coefficient of bimodality, used and described in Shade et al. (2014) and Ellison AM (1987).
• potential_analysis Repeats potential analysis (Livina et al. 2010) multiple times with bootstrap sampling for each row of the input data (as in Lahti et al. 2014) and returns the bootstrap score.

The coefficient lies in (0, 1).

The 'Sarle.asymptotic' version is defined as

\[ b = \frac{g^2 + 1}{k} \]
This is coefficient of bimodality from Ellison AM. Am. J. Bot. 1987, for microbiome analysis it has been used for instance in Shade et al. 2014. The formula for 'Sarle.finite.sample' (SAS 2012):

\[ b = \frac{g^2 + 1}{k + (3(n - 1)^2)/((n - 2)(n - 3))} \]

where \( n \) is sample size and \( k \) is the kth standardized moment (also called the sample kurtosis, or excess kurtosis).

**Value**

A list with following elements:

- score: Fraction of bootstrap samples where multiple modes are observed
- nmodes: The most frequently observed number of modes in bootstrap sampling results.
- results: Full results of potential_analysis for each row of the input matrix.

**Author(s)**

Leo Lahti <leo.lahti@iki.fi>

**References**

- Livina et al. (2010). Potential analysis reveals changing number of climate states during the last 60 kyr. *Climate of the Past*, 6, 77-82.
- To cite the microbiome R package, see citation('microbiome')

**See Also**

A classical test of multimodality is provided by dip.test in the DIP package.

**Examples**

```r
# In practice, use more bootstrap iterations
b <- bimodality(c(rnorm(100, mean=0), rnorm(100, mean=5)),
method = "Sarle.finite.sample", bs.iter=5)
# The classical DIP test:
# quantifies unimodality. Values range between 0 to 1.
# dip.test(x, simulate.p.value=TRUE, B=200)$statistic
# Values less than 0.05 indicate significant deviation from unimodality.
# Therefore, to obtain an increasing multimodality score, use
# library(diptest)
# multimodality.dip <- apply(abundances(pseq), 1,
# function (x) {1 - unname(dip.test(x)$p.value))}
```
Sarle’s Bimodality Coefficient

Description

Sarle’s bimodality coefficient.

Usage

bimodality_sarle(x, bs.iter = 1, type = "Sarle.finite.sample")

Arguments

x  
Data vector for which bimodality will be quantified

bs.iter  
Bootstrap iterations

type  
Score type (‘Sarle.finite.sample’ or ‘Sarle.asymptotic’)

Details

The coefficient lies in (0, 1).

The ‘Sarle.asymptotic’ version is defined as

\[ b = \frac{g^2 + 1}{k} \]

where \( g \) is sample skewness and \( k \) is the kth standardized moment (also called the sample kurtosis, or excess kurtosis).

The formula for ‘Sarle.finite.sample’ (SAS 2012):

\[ b = \frac{g^2 + 1}{k + (3(n - 1)^2)/(n - 2)(n - 3)} \]

where \( n \) is sample size

In both formulas, \( g \) is sample skewness and \( k \) is the kth standardized moment (also called the sample kurtosis, or excess kurtosis).

Value

Bimodality score

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>
boxplot_abundance

References

- To cite the microbiome R package, see citation('microbiome')

See Also

Check the dip.test from the DIP package for a classical test of multimodality.

Examples

```r
# b <- bimodality_sarle(rnorm(50), type='Sarle.finite.sample')
```

Description

Plot phyloseq abundances.

Usage

```r
boxplot_abundance(
  d,
  x,
  y,
  line = NULL,
  violin = FALSE,
  na.rm = FALSE,
  show.points = TRUE
)
```

Arguments

d  phyloseq-class object
x  Metadata variable to map to the horizontal axis.
y  OTU to map on the vertical axis
line  The variable to map on lines
violin  Use violin version of the boxplot
na.rm  Remove NAs
show.points  Include data points in the figure
boxplot_alpha

Details

The directionality of change in paired boxplot is indicated by the colors of the connecting lines.

Value

A `ggplot` plot object

Examples

data(peerj32)
p <- boxplot_abundance(peerj32$phyloseq, x='time', y='Akkermansia', line='subject')

boxplot_alpha             Alpha Boxplot

Description

Plot alpha index.

Usage

`boxplot_alpha(`

`x,`
`x_var = NULL,`
`index = NULL,`
`violin = FALSE,`
`na.rm = FALSE,`
`show.points = TRUE,`
`zeroes = TRUE,`
`element.alpha = 0.5,`
`element.width = 0.2,`
`fill.colors = NA,`
`outlier.fill = "grey50"`
```

Arguments

`x`       phyloseq-class object
`x_var`   Metadata variable to map to the horizontal axis.
`index`  Alpha index to plot. See function `alpha`.
`violin` Use violin version of the boxplot
`na.rm`  Remove NAs
`show.points` Include data points in the figure
`zeroes` Include zero counts in diversity estimation. Default is TRUE
element.alpha  Alpha value for plot elements. Controls the transparency of plots elements.

element.width  Width value for plot elements. Controls the transparency of plots elements.

fill.colors  Specify a list of colors passed on to ggplot2 scale_fill_manual

outlier.fill  If using boxplot and and points together how to deal with outliers. See ggplot2 outlier.fill argument in geom_elements.

Details

A simple wrapper to visualize alpha diversity index.

Value

A ggplot plot object

Examples

data("dietswap")
p <- boxplot_alpha(dietswap, x_var = "sex", index="observed", violin=FALSE,
  na.rm=FALSE, show.points=TRUE, zeroes=TRUE,
  element.alpha=0.5, element.width=0.2,
  fill.colors= c("steelblue", "firebrick"),
  outlier.fill="white")
p

chunk_reorder  Chunk Reorder

Description

Chunk re-order a vector so that specified newstart is first. Different than relevel.

Usage

chunk_reorder(x, newstart = x[[1]])

Details

Borrowed from phyloseq package as needed here and not exported there. Rewritten.

Value

Reordered x
Examples

# Typical use-case
# chunk_reorder(1:10, 5)
# # Default is to not modify the vector
# chunk_reorder(1:10)
# # Another example not starting at 1
# chunk_reorder(10:25, 22)
# # Should silently ignore the second element of `newstart`
# chunk_reorder(10:25, c(22, 11))
# # Should be able to handle `newstart` being the first argument already
# # without duplicating the first element at the end of `x`
# chunk_reorder(10:25, 10)
# # This is also the default
# all(chunk_reorder(10:25, 10) == 10:25)
# # An example with characters
# chunk_reorder(LETTERS, 'G')
# chunk_reorder(LETTERS, 'B')
# chunk_reorder(LETTERS, 'Z')
# # What about when `newstart` is not in `x`? Return x as-is, throw warning.
# chunk_reorder(LETTERS, 'g')

---

cmat2table Convert Correlation Matrix into a Table

Description

Arrange correlation matrices from associate into a table format.

Usage

cmat2table(res, verbose = FALSE)

Arguments

res Output from associate
verbose verbose

Value

Correlation table

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')
Examples

data(peerj32)
d1 <- peerj32$microbes[1:20, 1:10]
d2 <- peerj32$lipids[1:20, 1:10]
cc <- associate(d1, d2, mode='matrix', method='pearson')
cmat <- associate(d1, d2, mode='table', method='spearman')

collapse_replicates  Collapse Replicate Samples

Description

Collapse samples, mostly meant for technical replicates.

Usage

collapse_replicates(
  x,
  method = "sample",
  replicate_id = NULL,
  replicate_fields = NULL
)

Arguments

x  phyloseq-class object
method  Collapsing method. Only random sampling ("sample") implemented.
replicate_id  Replicate identifier. A character vector.
replicate_fields  Metadata fields used to determine replicates.

Value

Collapsed phyloseq object.

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

To cite the microbiome R package, see citation('microbiome')

Examples

data(atlas1006)
pseq <- collapse_replicates(atlas1006,
  method = "sample",
  replicate_fields = c("subject", "time"))
Description

Filter the phyloseq object to include only prevalent taxa.

Usage

core(x, detection, prevalence, include.lowest = FALSE, ...)

Arguments

- x phyloseq-class object
- detection Detection threshold for absence/presence (strictly greater by default).
- prevalence Prevalence threshold (in [0, 1]). The required prevalence is strictly greater by default. To include the limit, set include.lowest to TRUE.
- include.lowest Include the lower boundary of the detection and prevalence cutoffs. FALSE by default.
- ... Arguments to pass.

Value

Filtered phyloseq object including only prevalent taxa

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

Salonen A, Salojarvi J, Lahti L, de Vos WM. The adult intestinal core microbiota is determined by analysis depth and health status. Clinical Microbiology and Infection 18(S4):16-20, 2012 To cite the microbiome R package, see citation('microbiome')

See Also

core_members, rare_members

Examples

data(dietswap)
# Detection threshold 0 (strictly greater by default);
# Prevalence threshold 50 percent (strictly greater by default)
pseq <- core(dietswap, 0, 50/100)
# Detection threshold 0 (strictly greater by default);
# Prevalence threshold exactly 100 percent; for this set
# include.lowest=TRUE, otherwise the required prevalence is
# strictly greater than 100
pseq <- core(dietswap, 0, 100/100, include.lowest = TRUE)

---

core_abundance  Core Abundance

Description

Calculates the community core abundance index.

Usage

```r
core_abundance(
  x,
  detection = 0.1/100,
  prevalence = 50/100,
  include.lowest = FALSE
)
```

Arguments

- `x`  
  phyloseq-class object
- `detection`  
  Detection threshold for absence/presence (strictly greater by default).
- `prevalence`  
  Prevalence threshold (in [0, 1]). The required prevalence is strictly greater by default. To include the limit, set include.lowest to TRUE.
- `include.lowest`  
  Include the lower boundary of the detection and prevalence cutoffs. FALSE by default.

Details

The core abundance index gives the relative proportion of the core species (in [0,1]). The core taxa are defined as those that exceed the given population prevalence threshold at the given detection level.

Value

A vector of core abundance indices

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

See Also

rarity
**core_heatmap**

**Examples**

```r
data(dietswap)
d <- core_abundance(dietswap, detection=0.1/100, prevalence=50/100)
```

**Description**

Core heatmap.

**Usage**

```r
core_heatmap(x, dets, cols, min.prev, taxa.order)
```

**Arguments**

- `x`: OTU matrix
- `dets`: A vector or a scalar indicating the number of intervals in \((0, \log_{10}(\text{max(data)}))\). The dets are calculated for relative abundancies.
- `cols`: colours for the heatmap
- `min.prev`: If minimum prevalence is set, then filter out those rows (taxa) and columns (dets) that never exceed this prevalence. This helps to zoom in on the actual core region of the heatmap.
- `taxa.order`: Ordering of the taxa.

**Value**

Used for its side effects

**Author(s)**

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

**References**

A Salonen et al. The adult intestinal core microbiota is determined by analysis depth and health status. Clinical Microbiology and Infection 18(S4):16 20, 2012. To cite the microbiome R package, see citation('microbiome')
core_matrix

Description

Creates the core matrix.

Usage

core_matrix(x, prevalences = seq(0.1, 1, , 1), detections = NULL)

Arguments

- **x**  phyloseq object or a taxa x samples abundance matrix
- **prevalences**  a vector of prevalence percentages in [0,1]
- **detections**  a vector of intensities around the data range

Value

Estimated core microbiota

Author(s)

Contact: Jarkko Salojarvi <microbiome-admin@googlegroups.com>

References

A Salonen et al. The adult intestinal core microbiota is determined by analysis depth and health status. Clinical Microbiology and Infection 18(S4):16 20, 2012. To cite the microbiome R package, see citation('microbiome')

Examples

```r
# Not exported
#data(peerj32)
#core <- core_matrix(peerj32$phyloseq)
```
Description

Determine members of the core microbiota with given abundance and prevalences

Usage

core_members(x, detection = 1/100, prevalence = 50/100, include.lowest = FALSE)

Arguments

x  phyloseq-class object
detection  Detection threshold for absence/presence (strictly greater by default).
prevalence  Prevalence threshold (in [0, 1]). The required prevalence is strictly greater by default. To include the limit, set include.lowest to TRUE.
include.lowest  Include the lower boundary of the detection and prevalence cutoffs. FALSE by default.

Details

For phyloseq object, lists taxa that are more prevalent with the given detection threshold. For matrix, lists columns that satisfy these criteria.

Value

Vector of core members

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

A Salonen et al. The adult intestinal core microbiota is determined by analysis depth and health status. Clinical Microbiology and Infection 18(S4):16 20, 2012. To cite the microbiome R package, see citation('microbiome')

Examples

data(dietswap)
# Detection threshold 1 (strictly greater by default);
# Note that the data (dietswap) is here in absolute counts
# (and not compositional, relative abundances)
# Prevalence threshold 50 percent (strictly greater by default)
a <- core_members(dietswap, 1, 50/100)
Coverage Index

Description

Community coverage index.

Usage

coverage(x, threshold = 0.5)

Arguments

x A species abundance vector, or matrix (taxa/features x samples) with the absolute count data (no relative abundances), or phyloseq-class object

threshold Indicates the fraction of the ecosystem to be occupied by the N most abundant species (N is returned by this function). If the detection argument is a vector, then a data.frame is returned, one column for each detection threshold.

Details

The coverage index gives the number of groups needed to have a given proportion of the ecosystem occupied (by default 0.5 ie 50

Value

A vector of coverage indices

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

See Also

dominance, alpha

Examples

data(dietswap)
d <- coverage(dietswap, threshold=0.5)
Default Colors

Description
Default colors for different variables.

Usage

default_colors(x, v = NULL)

Arguments

x Name of the variable type ("Phylum")
v Optional. Vector of elements to color.

Value

Named character vector of default colors

Author(s)
Leo Lahti <leo.lahti@iki.fi>

References
See citation("microbiome")

Examples

col <- default_colors("Phylum")

Density Plot

Description
Density visualization for data points overlaid on cross-plot.
Usage

densityplot(
  x,
  main = NULL,
  x.ticks = 10,
  rounding = 0,
  add.points = TRUE,
  col = "black",
  adjust = 1,
  size = 1,
  legend = FALSE,
  shading = TRUE,
  shading.low = "white",
  shading.high = "black",
  point.opacity = 0.75
)

Arguments

  x        Data matrix to plot. The first two columns will be visualized as a cross-plot.
  main     title text
  x.ticks  Number of ticks on the X axis
  rounding Rounding for X axis tick values
  add.points Plot the data points as well
  col      Color of the data points. NAs are marked with darkgray.
  adjust   Kernel width adjustment
  size     point size
  legend   plot legend TRUE/FALSE
  shading  Shading
  shading.low Color for shading low density regions
  shading.high Color for shading high density regions
  point.opacity Transparency-level for points

Value

ggplot2 object

Author(s)

  Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

  See citation('microbiome')
dietswap

Examples

# p <- densityplot(cbind(rnorm(100), rnorm(100)))

---

dietswap Diet Swap Data

Description

The diet swap data set represents a study with African and African American groups undergoing a two-week diet swap. For details, see dx.doi.org/10.1038/ncomms7342.

Usage

data(dietswap)

Format

The data set in phyloseq-class format.

Details

The data is also available for download from the Data Dryad repository http://datadryad.org/resource/doi:10.5061/dryad.1mn1n.

Value

Loads the data set in R.

Author(s)

Leo Lahti <microbiome-admin@googlegroups.com>

References

O'Keefe et al. Nature Communications 6:6342, 2015. dx.doi.org/10.1038/ncomms7342 To cite the microbiome R package, see citation('microbiome')
**divergence**  
*Divergence within a Sample Group*

**Description**
Quantify microbiota divergence (heterogeneity) within a given sample set with respect to a reference.

**Usage**
```
divergence(x, y, method = "bray")
```

**Arguments**
- `x`: phyloseq object or a vector
- `method`: dissimilarity method: any method available via phyloseq::distance function. Note that some methods ("jsd" and 'unifrac' for instance) do not work with the group divergence.

**Details**
Microbiota divergence (heterogeneity / spread) within a given sample set can be quantified by the average sample dissimilarity or beta diversity with respect to a given reference sample.

This measure is sensitive to sample size. Subsampling or bootstrapping can be applied to equalize sample sizes between comparisons.

**Value**
Vector with dissimilarities; one for each sample, quantifying the dissimilarity of the sample from the reference sample.

**Author(s)**
Leo Lahti <microbiome-admin@googlegroups.com>

**References**
To cite this R package, see citation('microbiome')

**See Also**
the `vegdist` function from the *vegan* package provides many standard beta diversity measures
Examples

```r
# Assess beta diversity among the African samples
# in a diet swap study (see \code{help(dietswap)} for references)
data(dietswap)
pseq <- subset_samples(dietswap, nationality == 'AFR')
reference <- apply(abundances(pseq), 1, median)
b <- divergence(pseq, reference, method = "bray")
```

---

### Description

Various community diversity indices.

### Usage

```r
diversity(x, index = "all", zeroes = TRUE)
```

### Arguments

- `x`: A species abundance vector, or matrix (taxa/features x samples) with the absolute count data (no relative abundances), or phyloseq-class object
- `index`: Diversity index. See details for options.
- `zeroes`: Include zero counts in the diversity estimation.

### Details

By default, returns all diversity indices. The available diversity indices include the following:

- **inverse_simpson** Inverse Simpson diversity: $1/\lambda$ where $\lambda = \sum(p^2)$ and $p$ are relative abundances.
- **gini_simpson** Gini-Simpson diversity $1 - \lambda$. This is also called Gibbs–Martin, or Blau index in sociology, psychology and management studies.
- **shannon** Shannon diversity ie entropy
- **fisher** Fisher alpha; as implemented in the vegan package
- **coverage** Number of species needed to cover 50% of the ecosystem. For other quantiles, apply the function coverage directly.

### Value

A vector of diversity indices

### Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>
References


Bulla L. An index of diversity and its associated diversity measure. Oikos 70:167–171, 1994


See Also
dominance, richness, evenness, rarity, alpha

Examples

data(dietswap)
d <- alpha(dietswap, 'shannon')

dominance | Dominance Index
---|---

Description

Calculates the community dominance index.

Usage

dominance(x, index = "all", rank = 1, relative = TRUE, aggregate = TRUE)

Arguments

x A species abundance vector, or matrix (taxa/features x samples) with the absolute count data (no relative abundances), or phyloseq-class object

index If the index is given, it will override the other parameters. See the details below for description and references of the standard dominance indices. By default, this function returns the Berger-Parker index, ie relative dominance at rank 1.

rank Optional. The rank of the dominant taxa to consider.

relative Use relative abundances (default: TRUE)

aggregate Aggregate (TRUE; default) the top members or not. If aggregate=TRUE, then the sum of relative abundances is returned. Otherwise the relative abundance is returned for the single taxa with the indicated rank.
The dominance index gives the abundance of the most abundant species. This has been used also in microbiomics context (Locey & Lennon (2016)). The following indices are provided:

- 'absolute' This is the most simple variant, giving the absolute abundance of the most abundant species (Magurran & McGill 2011). By default, this refers to the single most dominant species (rank=1) but it is possible to calculate the absolute dominance with rank n based on the abundances of top-n species by tuning the rank argument.

- 'relative' Relative abundance of the most abundant species. This is with rank=1 by default but can be calculated for other ranks.

- 'DBP' Berger–Parker index, a special case of relative dominance with rank 1; This also equals the inverse of true diversity of the infinite order.

- 'DMN' McNaughton's dominance. This is the sum of the relative abundance of the two most abundant taxa, or a special case of relative dominance with rank 2

- 'simpson' Simpson's index ($\sum(p^2)$) where p are relative abundances has an interpretation as a dominance measure. Also the version ($\sum(q * (q-1)) / S(S-1)$) based on absolute abundances q has been proposed by Simpson (1949) but not included here as it is not within [0,1] range, and it is highly correlated with the simpler Simpson dominance. Finally, it is also possible to calculated dominances up to an arbitrary rank by setting the rank argument

- 'core_abundance' Relative proportion of the core species that exceed detection level 0.2% in over 50% of the samples

- 'gini' Gini index is calculated with the function inequality.

By setting aggregate=FALSE, the abundance for the single n'th most dominant taxa (n=rank) is returned instead the sum of abundances up to that rank (the default).

Value
A vector of dominance indices

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References


See Also
coverage, core_abundance, rarity, alpha
Examples

data(dietswap)
# vector
d <- dominance(abundances(dietswap)[,1], rank=1, relative=TRUE)
# matrix
# d <- dominance(abundances(dietswap), rank=1, relative=TRUE)
# Phyloseq object
# d <- dominance(dietswap, rank=1, relative=TRUE)

dominant  Dominant taxa

Description

Returns the dominant taxonomic group for each sample.

Usage

dominant(x, level = NULL)

Arguments

x  A phyloseq-class object
level  Optional. Taxonomic level.

Value

A vector of dominance indices

Author(s)

Leo Lahti <microbiome-admin@googlegroups.com>

Examples

data(dietswap)
# vector
d <- dominant(dietswap)
**estimate_stability**

Estimate Stability

**Description**

Quantify intermediate stability with respect to a given reference point.

**Usage**

```r
estimate_stability(df, reference.point = NULL, method = "lm", spl.list)
```

**Arguments**

- `df` Combined input data vector (samples x variables) and metadata data.frame (samples x features) with the 'data', 'subject' and 'time' field for each sample
- `reference.point` Optional. Calculate stability of the data w.r.t. this point. By default the intermediate range is used (min + (max - min)/2)
- `method` 'lm' (linear model) or 'correlation'; the linear model takes time into account as a covariate
- `spl` split object to speed up

**Details**

Decomposes each column in x into differences between consecutive time points. For each variable and time point we calculate for the data values: (i) the distance from reference point; (ii) distance from the data value at the consecutive time point. The 'correlation' method calculates correlation between these two variables. Negative correlations indicate that values closer to reference point tend to have larger shifts in the consecutive time point. The 'lm' method takes the time lag between the consecutive time points into account as this may affect the comparison and is not taken into account by the straightforward correlation. Here the coefficients of the following linear model are used to assess stability: abs(change) ~ time + abs(start.reference.distance). Samples with missing data, and subjects with less than two time point are excluded.

**Value**

A list with following elements: stability: estimated stability data: processed data set used in calculations

**Author(s)**

Leo Lahti <leo.lahti@iki.fi>
Examples

```r
# df <- data.frame(list(
#     subject=rep(paste('subject', 1:50, sep=' - '), each=2),
#     time=rep(1:2, 50),
#     data=rnorm(100)))
# s <- estimate_stability_single(df, reference.point=NULL, method='lm')
```

---

### evenness

**Evenness Index**

<table>
<thead>
<tr>
<th>evenness</th>
<th>Evenness Index</th>
</tr>
</thead>
</table>

**Description**

Various community evenness indices.

**Usage**

```r
evenness(x, index = "all", zeroes = TRUE, detection = 0)
```

**Arguments**

- `x` A species abundance vector, or matrix (taxa/features x samples) with the absolute count data (no relative abundances), or `phyloseq-class` object
- `index` Evenness index. See details for options.
- `zeroes` Include zero counts in the evenness estimation.
- `detection` Detection threshold

**Details**

By default, Pielou’s evenness is returned.

The available evenness indices include the following: 1) `camargo`: Camargo’s evenness (Camargo 1992) 2) `simpson`: Simpson’s evenness (inverse Simpson diversity / S) 3) `pielou`: Pielou’s evenness (Pielou, 1966), also known as Shannon or Shannon-Weaver/Wiener evenness; \( H/\ln(S) \). The Shannon-Weaver is the preferred term; see A tribute to Claude Shannon (1916 –2001) and a plea for more rigorous use of species richness, species diversity and the ‘Shannon–Wiener’ Index. Spellerberg and Fedor. Alpha Ecology & Biogeography (2003) 12, 177–197 4) `evar`: Smith and Wilson’s Evar index (Smith & Wilson 1996) 5) `bulla`: Bulla’s index (O) (Bulla 1994)

Desirable statistical evenness metrics avoid strong bias towards very large or very small abundances; are independent of richness; and range within \([0,1]\) with increasing evenness (Smith & Wilson 1996). Evenness metrics that fulfill these criteria include at least camargo, simpson, smith-wilson, and bulla. Also see Magurran & McGill (2011) and Beisel et al. (2003) for further details.

**Value**

A vector of evenness indices
find_optima

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References


See Also

coverage, core_abundance, rarity, alpha

Examples

```r
data(dietswap)
# phyloseq object
#d <- evenness(dietswap, 'pielou')
# matrix
#d <- evenness(abundances(dietswap), 'pielou')
# vector
d <- evenness(abundances(dietswap)[,1], 'pielou')
```

find_optima

Find Optima

Description

Detect optima, excluding local optima below peak.threshold.

Usage

```r
find_optima(f, peak.threshold = 0, bw = 1, min.density = 1)
```
Arguments

f  density
peak.threshold  Mode detection threshold
bw  bandwidth
min.density  Minimum accepted density for a maximum; as a multiple of kernel height

Value

A list with min (minima), max (maxima), and peak.threshold (minimum detection density)

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

See citation('microbiome')

Examples

# Not exported
# o <- find_optima(rnorm(100), bw=1)

Description

Measure association between nominal (no order for levels) variables

Usage

gktau(x, y)

Arguments

x  first variable
y  second variable
**Details**

Measure association between nominal (no order for levels) variables using Goodman and Kruskal tau. Code modified from the original source: r-bloggers.com/measuring-associations-between-non-numeric-variables/ An important feature of this procedure is that it allows missing values in either of the variables x or y, treating 'missing' as an additional level. In practice, this is sometimes very important since missing values in one variable may be strongly associated with either missing values in another variable or specific non-missing levels of that variable. An important characteristic of Goodman and Kruskal's tau measure is its asymmetry: because the variables x and y enter this expression differently, the value of $a(y,x)$ is not the same as the value of $a(x, y)$, in general. This stands in marked contrast to either the product-moment correlation coefficient or the Spearman rank correlation coefficient, which are both symmetric, giving the same association between x and y as that between y and x. The fundamental reason for the asymmetry of the general class of measures defined above is that they quantify the extent to which the variable x is useful in predicting y, which may be very different than the extent to which the variable y is useful in predicting x.

**Value**

Dependency measure

**Author(s)**

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

**References**

Code modified from the original source: http://r-bloggers.com/measuring-associations-between-non-numeric-variables/

To cite the microbiome R package, see citation('microbiome')

**Examples**

```r
data(peerj32)
v1 <- factor(peerj32$microbes[,1])
v2 <- factor(peerj32$meta$gender)
tc <- gktau(v1, v2)
```

---

<table>
<thead>
<tr>
<th>group_age</th>
<th>Age Classes</th>
</tr>
</thead>
</table>

**Description**

Cut age information to discrete factors.
Usage

```r
group_age(
  x,
  breaks = "decades",
  n = 10,
  labels = NULL,
  include.lowest = TRUE,
  right = FALSE,
  dig.lab = 3,
  ordered_result = FALSE
)
```

Arguments

- `x` Numeric vector (age in years)
- `breaks` Class break points. Either a vector of breakpoints, or one of the predefined options ("years", "decades", "even").
- `n` Number of groups for the breaks = "even" option.
- `labels` labels for the levels of the resulting category. By default, labels are constructed using "(a,b]" interval notation. If `labels = FALSE`, simple integer codes are returned instead of a factor.
- `include.lowest` logical, indicating if an ‘x[i]’ equal to the lowest (or highest, for `right = FALSE`) ‘breaks’ value should be included.
- `right` logical, indicating if the intervals should be closed on the right (and open on the left) or vice versa.
- `dig.lab` integer which is used when labels are not given. It determines the number of digits used in formatting the break numbers.
- `ordered_result` logical: should the result be an ordered factor?

Details

Regarding the breaks arguments, the "even" option aims to cut the samples in groups with approximately the same size (by quantiles). The "years" and "decades" options are self-explanatory.

Value

Factor of age groups.

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')
See Also
base::cut

Examples

data(atlas1006)
age.numeric <- meta(atlas1006)$age
age.factor <- group_age(age.numeric)

Description
Cut BMI information to standard discrete factors.

Usage

group_bmi(
  x,
  breaks = "standard",
  n = 10,
  labels = NULL,
  include.lowest = TRUE,
  right = FALSE,
  dig.lab = 3,
  ordered_result = FALSE
)

Arguments

  x             Numeric vector (BMI)
  breaks        Class break points. Either a vector of breakpoints, or one of the predefined
                 options ("standard", "standard_truncated", "even").
  n             Number of groups for the breaks = "even" option.
  labels        labels for the levels of the resulting category. By default, labels are constructed
                 using ",(a,b]" interval notation. If labels = FALSE, simple integer codes are
                 returned instead of a factor.
  include.lowest logical, indicating an ‘x[i]’ equal to the lowest (or highest, for right = FALSE)
                 ‘breaks’ value should be included.
  right         logical, indicating if the intervals should be closed on the right (and open on the
                 left) or vice versa.
  dig.lab       integer which is used when labels are not given. It determines the number of
                 digits used in formatting the break numbers.
  ordered_result logical: should the result be an ordered factor?
Details

Regarding the breaks arguments, the "even" option aims to cut the samples in groups with approximately the same size (by quantiles). The "standard" option corresponds to standard obesity categories defined by the cutoffs <18.5 (underweight); <25 (lean); <30 (obese); <35 (severe obese); <40 (morbid obese); <45 (super obese). The standard_truncated combines the severe, morbid and super obese into a single group.

Value

Factor of BMI groups.

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')

See Also

base::cut

Examples

bmi.numeric <- range(rnorm(100, mean = 25, sd = 3))
bmi.factor <- group_bmi(bmi.numeric)

heat

Association Heatmap

Description

Visualizes n x m association table as heatmap.

Usage

heat(
  df,
  Xvar = names(df)[[1]],
  Yvar = names(df)[[2]],
  fill = names(df)[[3]],
  star = NULL,
  p.adj.threshold = 1,
  association.threshold = 0,
  step = 0.2,
  colours = c("darkblue", "blue", "white", "red", "darkred"),
  limits = NULL,
heat

```r
legend.text = "",
order.rows = TRUE,
order.cols = TRUE,
filter.significant = TRUE,
star.size = NULL,
plot.values = FALSE
)
```

Arguments

- `df` Data frame. Each row corresponds to a pair of associated variables. The columns give variable names, association scores and significance estimates.
- `Xvar` X axis variable column name. For instance 'X'.
- `Yvar` Y axis variable column name. For instance 'Y'.
- `fill` Column to be used for heatmap coloring. For instance 'association'.
- `star` Column to be used for cell highlighting. For instance 'p.adj'.
- `p.adj.threshold` Significance threshold for the stars.
- `association.threshold` Include only elements that have absolute association higher than this value
- `step` color interval
- `colours` heatmap colours
- `limits` colour scale limits
- `legend.text` legend text
- `order.rows` Order rows to enhance visualization interpretability. If this is logical, then hclust is applied. If this is a vector then the rows are ordered using this index.
- `order.cols` Order columns to enhance visualization interpretability. If this is logical, then hclust is applied. If this is a vector then the rows are ordered using this index.
- `filter.significant` Keep only the elements with at least one significant entry
- `star.size` NULL Determine size of the highlight symbols
- `plot.values` Show values as text

Value

ggplot2 object

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')
Examples

data(peerj32)
d1 <- peerj32$lipids[, 1:10]
d2 <- peerj32$microbes[, 1:10]
cc <- associate(d1, d2, method='pearson')
p <- heat(cc, 'X1', 'X2', 'Correlation', star='p.adj')

Description

HITChip taxonomy table.

Usage

data(hitchip.taxonomy)

Format

List with the element 'filtered', including a simplified version of the HITChip taxonomy.

Value

Loads the data set in R.

Author(s)

Leo Lahti <microbiome-admin@googlegroups.com>

References

Lahti et al. Tipping elements of the human intestinal ecosystem. Nature Communications 5:4344, 2014. To cite the microbiome R package, see citation('microbiome')
**hotplot**

*Univariate Bimodality Plot*

**Description**

Coloured bimodality plot.

**Usage**

```r
hotplot(
  x, 
  taxon, 
  tipping.point = NULL, 
  lims = NULL, 
  shift = 0.001, 
  log10 = TRUE
)
```

**Arguments**

- `x` : *phyloseq-class* object
- `taxon` : Taxonomic group to visualize.
- `tipping.point` : Indicate critical point for abundance variations to be highlighted.
- `lims` : Optional. Figure X axis limits.
- `shift` : Small constant to avoid problems with zeroes in log10
- `log10` : Use log10 abundances for the OTU table and tipping point

**Value**

*ggplot* object

**Author(s)**

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

**References**

See citation('microbiome')

**Examples**

```r
data(atlas1006)
pseq <- subset_samples(atlas1006, DNA_extraction_method == 'r')
pseq <- transform(pseq, 'compositional')
# Set a tipping point manually
    tipp <- .3/100 # .3 percent relative abundance
# Bimodality is often best visible at log10 relative abundances
p <- hotplot(pseq, 'Dialister', tipping.point=tipp, log10=TRUE)
```
inequality

\textbf{Description}

Calculate Gini indices for a phyloseq object.

\textbf{Usage}

\texttt{inequality(x)}

\textbf{Arguments}

\texttt{x} \hspace{1cm} \textbf{phyloseq-class object}

\textbf{Details}

Gini index is a common measure for relative inequality in economical income, but can also be used as a community diversity measure. Gini index is between \([0,1]\), and increasing gini index implies increasing inequality.

\textbf{Value}

A vector of Gini indices

\textbf{Author(s)}

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

\textbf{References}


\textbf{See Also}

diversity, reldist::gini (inspired by that implementation but independently written here to avoid external dependencies)

\textbf{Examples}

\texttt{data(dietswap)}
\texttt{d \leftarrow inequality(dietswap)}
**intermediate_stability**

**Intermediate Stability**

**Description**

Quantify intermediate stability with respect to a given reference point.

**Usage**

```r
intermediate_stability(
  x,
  reference.point = NULL,
  method = "correlation",
  output = "scores"
)
```

**Arguments**

- **x**: phyloseq object. Includes abundances (variables x samples) and sample_data data.frame (samples x features) with 'subject' and 'time' field for each sample.
- **reference.point**: Calculate stability of the data w.r.t. this point. By default the intermediate range is used (min + (max - min)/2). If a vector of points is provided, then the scores will be calculated for every point and a data.frame is returned.
- **method**: 'lm' (linear model) or 'correlation'; the linear model takes time into account as a covariate
- **output**: Specify the return mode. Either the 'full' set of stability analysis outputs, or the 'scores' of intermediate stability.

**Details**

Decomposes each column in x into differences between consecutive time points. For each variable and time point we calculate for the data values: (i) the distance from reference point; (ii) distance from the data value at the consecutive time point. The 'correlation' method calculates correlation between these two variables. Negative correlations indicate that values closer to reference point tend to have larger shifts in the consecutive time point. The 'lm' method takes the time lag between the consecutive time points into account as this may affect the comparison and is not taken into account by the straightforward correlation. Here the coefficients of the following linear model are used to assess stability: abs(change) ~ time + abs(start.reference.distance). Samples with missing data, and subjects with less than two time point are excluded. The absolute count data x is logarithmized before the analysis with the log10(1 + x) trick to circumvent logarithmization of zeroes.

**Value**

A list with following elements: stability: estimated stability data; processed data set used in calculations
is_compositional

Author(s)
Leo Lahti <leo.lahti@iki.fi>

Examples

data(atlas1006)
x <- subset_samples(atlas1006, DNA_extraction_method == 'r')
x <- prune_taxa(c('Akkermansia', 'Dialister'), x)
res <- intermediate_stability(x, reference.point=NULL)

is_compositional Test Compositionality

Description
Test if phyloseq object is compositional.

Usage
is_compositional(x, tolerance = 1e-06)

Arguments
x phylrseq-class object
tolerance Tolerance for detecting compositionality.

Details
This function tests that the sum of abundances within each sample is almost zero, within the toler-
ance of 1e-6 by default.

Value
Logical TRUE/FALSE

See Also
transform

Examples

data(dietswap)
a <- is_compositional(dietswap)
b <- is_compositional(transform(dietswap, "identity"))
c <- is_compositional(transform(dietswap, "compositional"))
Description

Calculates the community rarity index by log-modulo skewness.

Usage

```r
log_modulo_skewness(x, q = 0.5, n = 50)
```

Arguments

- `x`: Abundance matrix (taxa x samples) with counts
- `q`: Arithmetic abundance classes are evenly cut up to this quantile of the data. The assumption is that abundances higher than this are not common, and they are classified in their own group.
- `n`: The number of arithmetic abundance classes from zero to the quantile cutoff indicated by `q`.

Details

The rarity index characterizes the concentration of species at low abundance. Here, we use the skewness of the frequency distribution of arithmetic abundance classes (see Magurran & McGill 2011). These are typically right-skewed; to avoid taking log of occasional negative skews, we follow Locey & Lennon (2016) and use the log-modulo transformation that adds a value of one to each measure of skewness to allow logarithmization.

Value

A vector of rarity indices

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References


See Also

core_abundance, low_abundance, alpha
low_abundance

Examples

```r
data(dietswap)
d <- log_modulo_skewness(dietswap)
```

---

**low_abundance**

*Low Abundance Index*

Description

Calculates the concentration of low-abundance taxa below the indicated detection threshold.

Usage

```r
low_abundance(x, detection = 0.2/100)
```

Arguments

- `x` phyloseq-class object
- `detection` Detection threshold for absence/presence (strictly greater by default).

Details

The `low_abundance` index gives the concentration of species at low abundance, or the relative proportion of rare species in [0,1]. The species that are below the indicated detection threshold are considered rare. Note that population prevalence is not considered. If the detection argument is a vector, then a data.frame is returned, one column for each detection threshold.

Value

A vector of indicators.

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

See Also

- core_abundance, rarity, global

Examples

```r
data(dietswap)
d <- low_abundance(dietswap, detection=0.2/100)
```
map_levels  

Map Taxonomic Levels

Description

Map taxa between hierarchy levels.

Usage

map_levels(taxa = NULL, from, to, data)

Arguments

taxa  
taxa to convert; if NULL then considering all taxa in the tax.table
from  
convert from taxonomic level

to  
convert to taxonomic level

data  
Either a phyloseq object or its taxonomyTable-class, see the phyloseq package.

Value

mappings

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')

Examples

data(dietswap)
m <- map_levels('Akkermansia', from='Genus', to='Phylum', tax_table(dietswap))
m <- map_levels('Verrucomicrobia', from='Phylum', to='Genus', tax_table(dietswap))
merge_taxa2  
Merge Taxa

Description

Merge taxonomic groups into a single group.

Usage

merge_taxa2(x, taxa = NULL, pattern = NULL, name = "Merged")

Arguments

x  phyloseq-class object
taxa   A vector of taxa names to merge.
pattern Taxa that match this pattern will be merged.
name   Name of the merged group.

Details

In some cases it is necessary to place certain OTUs or other groups into an "other" category. For instance, unclassified groups. This wrapper makes this easy. This function differs from phyloseq::merge_taxa by the last two arguments. Here, in merge_taxa2 the user can specify the name of the new merged group. And the merging can be done based on common pattern in the name.

Value

Modified phyloseq object

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')

Examples

data(dietswap)
s <- merge_taxa(dietswap, c())
Retrieves phyloseq metadata as a data frame.

**Description**

The output of the `phyloseq::sample_data()` function does not return data.frame, which is needed for many applications. This function retrieves the sample data as a data.frame.

**Usage**

```r
meta(x)
```

**Arguments**

- `x`: A phyloseq object.

**Value**

Sample metadata as a data.frame.

**Author(s)**

Leo Lahti <leo.lahti@iki.fi>

**See Also**

`sample_data` in the `phyloseq` package.

**Examples**

```r
data(dietswap); df <- meta(dietswap)
```

---

Multimodality score based on bootstrapped potential analysis.

**Usage**

```r
multimodality(
  x,
  peak.threshold = 1,
  bw.adjust = 1,
  bs.iter = 100,
  min.density = 1,
  verbose = TRUE
)
```
Arguments

x                A vector, or data matrix (variables x samples)
peak.threshold  Mode detection threshold
bw.adjust       Bandwidth adjustment
bs.iter         Bootstrap iterations
min.density      minimum accepted density for a maximum; as a multiple of kernel height
verbose         Verbose

Details

Repeats potential analysis (Livina et al. 2010) multiple times with bootstrap sampling for each row of the input data (as in Lahti et al. 2014) and returns the specified results.

Value

A list with following elements:

• score Fraction of bootstrap samples with multiple observed modes
• nmodes The most frequently observed number of modes in bootstrap
• results Full results of potential_analysis for each row of the input matrix.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

• Livina et al. (2010). Potential analysis reveals changing number of climate states during the last 60 kyr. Climate of the Past, 6, 77-82.

Examples

#data(peerj32)
#s <- multimodality(t(peerj32$microbes[, c('Akkermansia', 'Dialister')]))
**neat**  

*Neatmap Sorting*

**Description**

Order matrix or phyloseq OTU table based on the neatmap approach.

**Usage**

```r
neat(
  x,
  arrange = "both",
  method = "NMDS",
  distance = "bray",
  first.feature = NULL,
  first.sample = NULL,
  ...
)
```

**Arguments**

- `x`: A matrix or phyloseq object.
- `arrange`: Order 'features', 'samples' or 'both' (for matrices). For matrices, it is assumed that the samples are on the columns and features are on the rows. For phyloseq objects, features are the taxa of the OTU table.
- `method`: Ordination method. Only NMDS implemented for now.
- `distance`: Distance method. See `vegdist` function from the `vegan` package.
- `first.feature`: Optionally provide the name of the first feature to start the ordering
- `first.sample`: Optionally provide the name of the first sample to start the ordering
- `...`: Arguments to pass.

**Details**

Borrows elements from the heatmap implementation in the `phyloseq` package. The row/column sorting is not available there as a separate function. Therefore I implemented this function to provide an independent method for easy sample/taxon reordering for phyloseq objects. The ordering is cyclic so we can start at any point. The choice of the first sample may somewhat affect the overall ordering.

**Value**

Sorted matrix
References

This function is partially based on code derived from the phyloseq package. However for the original neatmap approach for heatmap sorting, see (and cite): Rajaram, S., & Oono, Y. (2010). NeatMap—non-clustering heat map alternatives in R. BMC Bioinformatics, 11, 45.

Examples

data(peerj32)
# Take subset to speed up example
x <- peerj32$microbes[1:10,1:10]
oxo <- neat(x, 'both', method='NMDS', distance='bray')

neatsort

Neatmap Sorting

Description

Sort samples or features based on the neatmap approach.

Usage

neatsort(x, target, method = "NMDS", distance = "bray", first = NULL, ...)

Arguments

x phyloseq-class object or a matrix
target For phyloseq-class input, the target is either 'sites' (samples) or 'species' (features) (taxa/OTUs); for matrices, the target is 'rows' or 'cols'.
method Ordination method. See ordinate from phyloseq package. For matrices, only the NMDS method is available.
distance Distance method. See ordinate from phyloseq package.
first Optionally provide the name of the first sample/taxon to start the ordering (the ordering is cyclic so we can start at any point). The choice of the first sample may somewhat affect the overall ordering.
... Arguments to be passed.

Details

This function borrows elements from the heatmap implementation in the phyloseq package. The row/column sorting is there not available as a separate function at present, however, hindering reuse in other tools. Implemented in the microbiome package to provide an independent method for easy sample/taxon reordering for phyloseq objects.

Value

Vector of ordered elements
References

This function is partially based on code derived from the `phyloseq` package. For the original neatmap approach for heatmap sorting, see (and cite): Rajaram, S., & Oono, Y. (2010). NeatMap—non-clustering heat map alternatives in R. BMC Bioinformatics, 11, 45.

Examples

```r
data(peerj32)
pseq <- peerj32$phyloseq
# For Phyloseq
sort.otu <- neatsort(pseq, target='species')
# For matrix
# sort.rows <- neatsort(abundances(pseq), target='rows')
```

---

### overlap

<table>
<thead>
<tr>
<th>Overlap Measure</th>
</tr>
</thead>
</table>

#### Description

Quantify microbiota 'overlap' between samples.

#### Usage

```r
overlap(x, detection = 0)
```

#### Arguments

- `x` 
  - `phyloseq-class` object
- `detection` 
  - Detection threshold.

#### Value

Overlap matrix

#### Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

#### References


#### Examples

```r
data(atlas1006)
o <- overlap(atlas1006, detection = 0.1/100)
```
Description

The peerj32 data set contains high-through profiling data from 389 human blood serum lipids and 130 intestinal genus-level bacteria from 44 samples (22 subjects from 2 time points; before and after probiotic/placebo intervention). The data set can be used to investigate associations between intestinal bacteria and host lipid metabolism. For details, see http://dx.doi.org/10.7717/peerj.32.

Usage

data(peerj32)

Format

List of the following data matrices as described in detail in Lahti et al. (2013):

- lipids: Quantification of 389 blood serum lipids across 44 samples
- microbes: Quantification of 130 genus-like taxa across 44 samples
- meta: Sample metadata including time point, sex, subjectID, sampleID and treatment group (probiotic LGG / Placebo)
- phyloseq The microbiome data set converted into a phyloseq-class object.

Value

Loads the data set in R.

Author(s)

Leo Lahti <microbiome-admin@googlegroups.com>

References

Lahti et al. (2013) PeerJ 1:e32 http://dx.doi.org/10.7717/peerj.32
plot_atlas  Visualize Samples of a Microbiota Atlas

Description

Show all samples of a microbiota collection, colored by specific factor levels (x axis) and signal (y axis).

Usage

plot_atlas(pseq, x, y, ncol = 2)

Arguments

pseq  phyloseq object
x  Sorting variable for X axis and sample coloring
y  Signal variable for Y axis
ncol  Number of legend columns.

Details

Arranges the samples based on the given grouping factor (x), and plots the signal (y) on the Y axis. The samples are randomly ordered within each factor level. The factor levels are ordered by standard deviation of the signal (y axis).

Value

ggplot object

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

See citation('microbiome'); Visualization inspired by Kilpinen et al. 2008, Genome Biology 9:R139. DOI: 10.1186/gb-2008-9-9-r139

Examples

data(atlas1006)
p <- plot_atlas(atlas1006, 'DNA_extraction_method', 'diversity')p <- plot_atlas(atlas1006, 'DNA_extraction_method', 'Bifidobacterium')
plot_composition

Taxonomic Composition Plot

Description
Plot taxon abundance for samples.

Usage
plot_composition(
  x,
  sample.sort = NULL,
  otu.sort = NULL,
  x.label = "sample",
  plot.type = "barplot",
  verbose = FALSE,
  average_by = NULL,
  group_by = NULL,
  ...
)

Arguments
x          phyloseq-class object
sample.sort Order samples. Various criteria are available:
  * NULL or 'none': No sorting
  * A single character string: indicate the metadata field to be used for ordering.
    Or: if this string is found from the tax_table, then sort by the corresponding
    taxonomic group.
  * A character vector: sample IDs indicating the sample ordering.
  * 'neatmap' Order samples based on the neatmap approach. See neatsort. By default, 'NMDS'
    method with 'bray' distance is used. For other options, arrange the samples manually with the function.

otu.sort   Order taxa. Same options as for the sample.sort argument but instead of meta-
            data, taxonomic table is used. Also possible to sort by 'abundance'.

x.label    Specify how to label the x axis. This should be one of the variables in sam-
            ple_variables(x).

plot.type  Plot type: 'barplot' or 'heatmap'

verbose    verbose (but not in sample/taxon ordering). The options are 'Z-OTU', 'Z-Sample',
            'log10' and 'compositional'. See the transform function.

average_by Average the samples by the average_by variable

group_by   Group by this variable (in plot.type "barplot")

...         Arguments to be passed (for neatsort function)
plot_core

Value

A `ggplot` plot object.

Examples

```r
library(dplyr)
data(atlas1006)
pseq <- atlas1006 %>%
  subset_samples(DNA_extraction_method == "r") %>%
  aggregate_taxa(level = "Phylum") %>%
  transform(transform = "compositional")
p <- plot_composition(pseq, sample.sort = "Firmicutes",
  otu.sort = "abundance", verbose = TRUE) +
  scale_fill_manual(values = default_colors("Phylum")[taxa(pseq)])
```

plot_core

Visualize OTU Core

Description

Core visualization (2D).

Usage

```r
plot_core(
  x,
  prevalences = seq(0.1, 1, 0.1),
  detections = 20,
  plot.type = "lineplot",
  colours = NULL,
  min.prevalence = NULL,
  taxa.order = NULL,
  horizontal = FALSE
)
```

Arguments

- **x**
  A `phyloseq` object or a core matrix
- **prevalences**
  a vector of prevalence percentages in [0,1]
- **detections**
  a vector of intensities around the data range, or a scalar indicating the number of intervals in the data range.
- **plot.type**
  Plot type ('lineplot' or 'heatmap')
- **colours**
  colours for the heatmap
- **min.prevalence**
  If minimum prevalence is set, then filter out those rows (taxa) and columns (detections) that never exceed this prevalence. This helps to zoom in on the actual core region of the heatmap. Only affects the plot.type='heatmap'.
- **taxa.order**
  Ordering of the taxa: a vector of names.
- **horizontal**
  Logical. Horizontal figure.
Value

A list with three elements: the ggplot object and the data. The data has a different form for the lineplot and heatmap. Finally, the applied parameters are returned.

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

A Salonen et al. The adult intestinal core microbiota is determined by analysis depth and health status. Clinical Microbiology and Infection 18(S4):16 20, 2012. To cite the microbiome R package, see citation('microbiome')

Examples

data(dietswap)
p <- plot_core(transform(dietswap, "compositional"),
prevalences=seq(0.1, 1, .1), detections=seq(0.01, 1, length = 10))
plot_frequencies

Arguments

x  phyloseq-class object or an OTU matrix (samples x phylotypes)
variable  OTU or metadata variable to visualize
log10  Logical. Show log10 abundances or not.
adjust  see stat_density
kernel  see stat_density
trim  see stat_density
na.rm  see stat_density
fill  Fill color
tipping.point  Optional. Indicate critical point for abundance variations to be highlighted.
xlim  X axis limits

Value

A ggplot plot object.

Examples

# Load gut microbiota data on 1006 western adults
# (see help(atlas1006) for references and details)
data(dietswap)
# Use compositional abundances instead of absolute signal
pseq.rel <- transform(dietswap, 'compositional')
# Population density for Dialister spp.; with log10 on the abundance (X)
# axis
library(ggplot2)
p <- plot_density(pseq.rel, variable='Dialister') + scale_x_log10()

Description

Plot relative frequencies within each Group for the levels of the given factor.

Usage

plot_frequencies(x, Groups, Factor)

Arguments

x  data.frame
Groups  Name of the grouping variable
Factor  Name of the frequency variable
Details

For table with the indicated frequencies, see the returned phyloseq object.

Value

*ggplot* plot object.

Examples

```r
data(dietswap)
p <- plot_frequencies(meta(dietswap), 'group', 'sex')
```

### plot_landscape

**Landscape Plot**

Description

Wrapper for visualizing sample similarity landscape ie. sample density in various 2D projections.

Usage

```r
plot_landscape(
  x, method = "PCoA", distance = "bray", transformation = "identity",
  col = NULL, main = NULL, x.ticks = 10, rounding = 0, add.points = TRUE,
  adjust = 1, size = 1, legend = FALSE, shading = TRUE,
  shading.low = "#ebf4f5", shading.high = "#e9b7ce",
  point.opacity = 0.75
)
```

Arguments

- **x**  
  *phyloseq-class* object or a data matrix (samples x features; eg. samples vs. OTUs). If the input x is a 2D matrix then it is plotted as is.

- **method**  
  Ordination method, see phyloseq:plot_ordination; or "PCA", or "t-SNE" (from the Rtsne package)
distance Ordination distance, see phyloseq::plot_ordination; for method = "PCA", only euclidean distance is implemented now.

transformation Transformation applied on the input object x

col Variable name to highlight samples (points) with colors

main title text

x.ticks Number of ticks on the X axis

rounding Rounding for X axis tick values

add.points Plot the data points as well

adjust Kernel width adjustment

size point size

legend plot legend TRUE/FALSE

shading Add shading in the background.

shading.low Color for shading low density regions

shading.high Color for shading high density regions

point.opacity Transparency-level for points

Details

For consistent results, set random seed (set.seed) before function call. Note that the distance and transformation arguments may have a drastic effect on the outputs.

Value

A ggplot plot object.

Examples

data(dietswap)

# PCoA
p <- plot_landscape(transform(dietswap, "compositional"),
                     distance = "bray", method = "PCoA")

p <- plot_landscape(dietswap, method = "t-SNE", distance = "bray",
                     transformation = "compositional")

# PCA
p <- plot_landscape(dietswap, method = "PCA", transformation = "clr")
plot_regression  Visually Weighted Regression Plot

Description


Usage

plot_regression(
  formula,  # formula
  data,     # data
  B = 1000,  # number bootstrapped smoothers
  shade = TRUE,  # plot the shaded confidence region?
  shade.alpha = 0.1,  # shade.alpha: should the CI shading fade out at the edges? (by reducing alpha; 0=no alpha decrease, 0.1=medium alpha decrease, 0.5=strong alpha decrease)
  spag = FALSE,  # plot spaghetti lines?
  mweight = TRUE,  # visually weight the median smoother
  show.lm = FALSE,  # plot the linear regression line
  show.median = TRUE,  # show the median smoother
  median.col = "white",  # median.col
  show.CI = FALSE,  # show confidence intervals
  method = loess,  # loess
  slices = 200,  # slices
  ylim = NULL,  # ylim
  quantize = "continuous",  # quantize
  show.points = TRUE,  # show points
  color = NULL,  # color
  pointsize = NULL,  # pointsize
  ...  # ...
)

Arguments

formula            formula
data               data
B                   number bootstrapped smoothers
shade               plot the shaded confidence region?
shade.alpha         shade.alpha: should the CI shading fade out at the edges? (by reducing alpha; 0=no alpha decrease, 0.1=medium alpha decrease, 0.5=strong alpha decrease)
spag                plot spaghetti lines?
mweight             visually weight the median smoother
show.lm             plot the linear regression line
show.median show median smoother
median.col median color
show.CI should the 95% CI limits be plotted?
method the fitting function for the spaghettis; default: loess
slices number of slices in x and y direction for the shaded region. Higher numbers make a smoother plot, but takes longer to draw. I wouldn'T go beyond 500
ylim restrict range of the watercoloring
quantize either 'continuous', or 'SD'. In the latter case, we get three color regions for 1, 2, and 3 SD (an idea of John Mashey)
show.points Show points.
color Point colors
pointsize Point sizes
... further parameters passed to the fitting function, in the case of loess, for example, 'span=.9', or 'family='symmetric"

Value

ggplot2 object

Author(s)

Based on the original version from F. Schonbrodt. Modified by Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')

Examples

data(atlas1006)
pseq <- subset_samples(atlas1006,
  DNA_extraction_method == 'r' &
  sex == "female" &
  nationality == "UKIE",
  B=10, slices=10 # non-default used here to speed up examples
)
p <- plot_regression(diversity ~ age, meta(pseq)[1:20,], slices=10, B=10)
plot_taxa_prevalence  
*Visualize Prevalence Distributions for Taxa*

**Description**
Create taxa prevalence plots at various taxonomic levels.

**Usage**
```r
plot_taxa_prevalence(x, level, detection = 0)
```

**Arguments**
- `x`  
  *phyloseq-class* object, OTU data must be counts and not relative abundance or other transformed data.
- `level`  
  Phylum/Order/Class/Family
- `detection`  
  Detection threshold for presence (prevalence)

**Details**
This helps to obtain first insights into how is the taxa distribution in the data. It also gives an idea about the taxonomic affiliation of rare and abundant taxa in the data. This may be helpful for data filtering or other downstream analysis.

**Value**
A `ggplot` plot object.

**Author(s)**
Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

**Examples**
```r
data(atlas1006)
# Pick data subset just to speed up example
p0 <- subset_samples(atlas1006, DNA_extraction_method == "r")
p0 <- prune_taxa(taxa(p0)[grep("Bacteroides", taxa(p0))], p0)
# Detection threshold (0 by default; higher especially with HITChip)
p <- plot_taxa_prevalence(p0, 'Phylum', detection = 1)
print(p)
```
**plot_tipping**  
**Variation Line Plot**

**Description**
Plot variation in taxon abundance for many subjects.

**Usage**

```r
plot_tipping(
  x, 
  taxon, 
  tipping.point = NULL, 
  lims = NULL, 
  shift = 0.001, 
  xlim = NULL 
)
```

**Arguments**
- **x**  
  *phyloseq-class* object
- **taxon**  
  Taxonomic group to visualize.
- **tipping.point**  
  Optional. Indicate critical point for abundance variations to be highlighted.
- **lims**  
  Optional. Figure X axis limits.
- **shift**  
  Small constant to avoid problems with zeroes in log10
- **xlim**  
  Horizontal axis limits

**Details**
Assuming the sample_data(x) has 'subject' field and some subjects have multiple time points.

**Value**

`ggplot` object

**Author(s)**
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

**References**
See citation('microbiome')
Examples

data(atlas1006)
pseq <- subset_samples(atlas1006, DNA_extraction_method == 'r')
pseq <- transform(pseq, 'compositional')
p <- plot_tipping(pseq, 'Dialister', tipping.point=1)

potential_analysis  

Description

Analysis of multimodality based on bootstrapped potential analysis of Livina et al. (2010) as described in Lahti et al. (2014).

Usage

potential_analysis(
  x,
  peak.threshold = 0,
  bw.adjust = 1,
  bs.iter = 100,
  min.density = 1
)

Arguments

  x             Input data vector
  peak.threshold Mode detection threshold
  bw.adjust     Bandwidth adjustment
  bs.iter       Bootstrap iterations
  min.density   minimum accepted density for a maximum; as a multiple of kernel height

Value

List with following elements:

  • modes Number of modes for the input data vector (the most frequent number of modes from bootstrap)
  • minima Average of potential minima across the bootstrap samples (for the most frequent number of modes)
  • maxima Average of potential maxima across the bootstrap samples (for the most frequent number of modes)
  • unimodality.support Fraction of bootstrap samples exhibiting unimodality
  • bws Bandwidths
potential_univariate

References

• Livina et al. (2010). Potential analysis reveals changing number of climate states during the last 60 kyr. *Climate of the Past*, 6, 77-82.


See Also

plot_potential

Examples

# Example data; see help(peerj32) for details
data(peerj32)

# Log10 abundance of Dialister
x <- abundances(transform(peerj32$phyloseq, "clr"))['Dialister',]

# Bootstrapped potential analysis
# In practice, use more bootstrap iterations
# res <- potential_analysis(x, peak.threshold=0, bw.adjust=1,
#                           bs.iter=9, min.density=1)

potential_univariate

Potential Analysis for Univariate Data

Description

One-dimensional potential estimation for univariate timeseries.

Usage

potential_univariate(x, std = 1, bw = "nrd", weights = c(), grid.size = NULL, peak.threshold = 1, bw.adjust = 1, density.smoothing = 0, min.density = 1)
potential_univariate

Arguments

- **x**: Univariate data (vector) for which the potentials shall be estimated
- **std**: Standard deviation of the noise (defaults to 1; this will set scaled potentials)
- **bw**: kernel bandwidth estimation method
- **weights**: optional weights in ksdensity (used by potential_slidingaverages).
- **grid.size**: Grid size for potential estimation. of density kernel height \( \frac{\text{dnorm}(0, \text{sd}=\text{bandwidth})}{\text{N}} \)
- **peak.threshold**: Mode detection threshold
- **bw.adjust**: The real bandwidth will be \( \text{bw.adjust}*\text{bw} \); defaults to 1
- **density.smoothing**: Add a small constant density across the whole observation range to regularize density estimation (and to avoid zero probabilities within the observation range). This parameter adds uniform density across the observation range, scaled by density.smoothing.
- **min.density**: minimum accepted density for a maximum; as a multiple of kernel height

Value

potential_univariate returns a list with the following elements:

- **xi**: the grid of points on which the potential is estimated
- **pot**: The estimated potential: \(-\log(f)\times\text{std}^2/2\), where \(f\) is the density.
- **density**: Density estimate corresponding to the potential.
- **min.inds**: indices of the grid points at which the density has minimum values; (-potentials; neglecting local optima)
- **max.inds**: indices the grid points at which the density has maximum values; (-potentials; neglecting local optima)
- **bw**: bandwidth of kernel used
- **min.points**: grid point values at which the density has minimum values; (-potentials; neglecting local optima)
- **max.points**: grid point values at which the density has maximum values; (-potentials; neglecting local optima)

Author(s)

Based on Matlab code from Egbert van Nes modified by Leo Lahti. Extended from the initial version in the earlywarnings R package.

References

- Livina et al. (2010). Potential analysis reveals changing number of climate states during the last 60 kyr. *Climate of the Past*, 6, 77-82.
Examples

```r
# res <- potential_univariate(x)
```

<table>
<thead>
<tr>
<th>prevalence</th>
<th>OTU Prevalence</th>
</tr>
</thead>
</table>

Description

Simple prevalence measure.

Usage

```r
prevalence(
  x,
  detection = 0,
  sort = FALSE,
  count = FALSE,
  include.lowest = FALSE
)
```

Arguments

- `x`: A vector, data matrix or `phyloseq` object
- `detection`: Detection threshold for absence/presence (strictly greater by default).
- `sort`: Sort the groups by prevalence
- `count`: Logical. Indicate prevalence as fraction of samples (in percentage \([0, 1]\); default); or in absolute counts indicating the number of samples where the OTU is detected (strictly) above the given abundance threshold.
- `include.lowest`: Include the lower boundary of the detection and prevalence cutoffs. FALSE by default.

Details

For vectors, calculates the fraction (count=FALSE) or number (count=TRUE) of samples that exceed the detection. For matrices, calculates this for each matrix column. For phyloseq objects, calculates this for each OTU. The relative prevalence (count=FALSE) is simply the absolute prevalence (count=TRUE) divided by the number of samples.

Value

For each OTU, the fraction of samples where a given OTU is detected. The output is readily given as a percentage.

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>
References

A Salonen et al. The adult intestinal core microbiota is determined by analysis depth and health status. Clinical Microbiology and Infection 18(S4):16 20, 2012. To cite the microbiome R package, see citation('microbiome')

Examples

data(peerj32)
pr <- prevalence(peerj32$phyloseq, detection=0, sort=TRUE, count=TRUE)

psmelt2

Convert phyloseq-class object to long data format

Description

An alternative to psmelt function from phyloseq-class object.

Usage

psmelt2(x, sample.column = NULL, feature.column = NULL)

Arguments

x phyloseq-class object

sample.column A single character string specifying name of the column to hold sample names.

feature.column A single character string specifying name of the column to hold OTU or ASV names.

Value

A tibble in long format

Author(s)

Contact: Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

Examples

data("dietswap")
ps.melt <- psmelt2(dietswap, sample.column="SampleID",
                    feature.column="Feature")
head(ps.melt)
**quiet**  

*Quiet Output*

**Description**

Suppress all output from an expression. Works cross-platform.

**Usage**

quiet(expr, all = TRUE)

**Arguments**

- `expr` Expression to run.
- `all` If TRUE then suppress warnings and messages as well; otherwise, only suppress printed output (such as from print or cat).

**Value**

Used for its side effects.

**Author(s)**

Adapted from https://gist.github.com/daattali/6ab55ae6b50e8929d89

**Examples**

quiet(1 + 1)

---

**radial_theta**  

*Radial Theta Function*

**Description**

Adapted from NeatMap and phyloseq packages but not exported and hence not available via phyloseq. Completely rewritten to avoid license conflicts. Vectorized to gain efficiency; only calculates theta and omits r.

**Usage**

radial_theta(x)

**Arguments**

- `x` position parameter
rare

Description
Filter the phyloseq object to include only rare (non-core) taxa.

Usage
rare(x, detection, prevalence, include.lowest = FALSE, ...)

Arguments
x phyloseq-class object
detection Detection threshold for absence/presence (strictly greater by default).
prevalence Prevalence threshold (in [0, 1]; strictly greater by default)
include.lowest Include the lower boundary of the detection and prevalence cutoffs in core calculation. FALSE by default.
... Arguments to pass.

Value
Filtered phyloseq object including only rare taxa

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References
Salonen A, Salojarvi J, Lahti L, de Vos WM. The adult intestinal core microbiota is determined by analysis depth and health status. Clinical Microbiology and Infection 18(S4):16-20, 2012 To cite the microbiome R package, see citation('microbiome')

See Also
core_members

Examples
data(dietswap)
# Detection threshold 0 (strictly greater by default);
# Prevalence threshold 50 percent (strictly greater by default)
pseq <- rare(dietswap, 0, 50/100)
rare_abundance

Rare (Non-Core) Abundance Index

Description
Calculates the rare abundance community index.

Usage
rare_abundance(
  x,
  detection = 0.1/100,
  prevalence = 50/100,
  include.lowest = FALSE
)

Arguments
  x phyloseq-class object
  detection  Detection threshold for absence/presence (strictly greater by default).
  prevalence Prevalence threshold (in [0, 1]). The required prevalence is strictly greater by default. To include the limit, set include.lowest to TRUE.
  include.lowest Include the lower boundary of the detection and prevalence cutoffs. FALSE by default.

Details
This index gives the relative proportion of rare species (ie. those that are not part of the core microbiota) in the interval [0,1]. This is the complement (1-x) of the core abundance. The rarity function provides the abundance of the least abundant taxa within each sample, regardless of the population prevalence.

Value
A vector of indices

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

See Also
core_abundance, rarity, diversity

Examples
data(dietswap)
d <- rare_abundance(dietswap, detection=0.1/100, prevalence=50/100)
rare_members

Rare Taxa

Description
Determine members of the rare microbiota with given abundance and prevalence threshold.

Usage
rare_members(x, detection = 1/100, prevalence = 50/100, include.lowest = FALSE)

Arguments
- **x** phyloseq-class object
- **detection** Detection threshold for absence/presence (strictly greater by default).
- **prevalence** Prevalence threshold (in [0, 1]). The required prevalence is strictly greater by default. To include the limit, set include.lowest to TRUE.
- **include.lowest** Include the lower boundary of the detection and prevalence cutoffs. FALSE by default.

Details
For phyloseq object, lists taxa that are less prevalent than the given prevalence threshold. Optionally, never exceeds the given abundance threshold (by default, all abundances accepted). For matrix, lists columns that satisfy these criteria.

Value
Vector of rare taxa

Author(s)
Leo Lahti <microbiome-admin@googlegroups.com>

References
To cite the microbiome R package, see citation('microbiome')

See Also
core_members

Examples
data(dietswap)
# Detection threshold: the taxa never exceed the given detection threshold
# Prevalence threshold 20 percent (strictly greater by default)
a <- rare_members(dietswap, detection=100/100, prevalence=20/100)
**rarity**  
*Rarity Index*

**Description**
Calculates the community rarity index.

**Usage**

```r
rarity(x, index = "all", detection = 0.2/100, prevalence = 20/100)
```

**Arguments**

- `x` phyloseq-class object
- `index` If the index is given, it will override the other parameters. See the details below for description and references of the standard rarity indices.
- `detection` Detection threshold for absence/presence (strictly greater by default).
- `prevalence` Prevalence threshold (in [0, 1]). The required prevalence is strictly greater by default. To include the limit, set include.lowest to TRUE.

**Details**

The rarity index characterizes the concentration of species at low abundance.

The following rarity indices are provided:

- **log_modulo_skewness** Quantifies the concentration of the least abundant species by the log-modulo skewness of the arithmetic abundance classes (see Magurran & McGill 2011). These are typically right-skewed; to avoid taking log of occasional negative skews, we follow Locey & Lennon (2016) and use the log-modulo transformation that adds a value of one to each measure of skewness to allow logarithmization. The values q=0.5 and n=50 are used here.
- **low_abundance** Relative proportion of the least abundant species, below the detection level of 0.2%. The least abundant species are determined separately for each sample regardless of their prevalence.
- **rare_abundance** Relative proportion of the non-core species, exceed the given detection level (default 20 at the given prevalence (default 20). This is complement of the core with the same thresholds.

**Value**
A vector of rarity indices

**Author(s)**
Contact: Leo Lahti <microbiome-admin@googlegroups.com>
References


See Also

alpha, log_modulo_skewness, rare_abundance, low_abundance

Examples

```r
data(dietswap)
d <- rarity(dietswap, index='low_abundance')
# d <- rarity(dietswap, index='all')
```

---

### readcount

#### Description

Total Read Count

#### Usage

```r
readcount(x)
```

#### Arguments

- `x`  
  *phyloseq-class* object

#### Value

Vector of read counts.

#### Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

#### References

See citation('microbiome')

#### Examples

```r
data(dietswap)
d <- readcount(dietswap)
```
Description

Read biom and mapping files into a phyloseq-class object.

Usage

read_biom2phyloseq(
  biom.file = NULL,
  taxonomy.file = NULL,
  metadata.file = NULL,
  ...
)

Arguments

  biom.file # A biom file with '.biom' extension
  taxonomy.file # NULL the latest version has taxonomic information within the biom
  metadata.file # A simple metadata/mapping file with .csv extension
  ... # Arguments to pass for import_biom

Details

Biom file and mapping files will be converted to phyloseq-class.

Value

phyloseq-class object.

Author(s)

Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

Examples

p0 <- read_biom2phyloseq()
#biom.file <- qiita1629.biom"
#meta.file <- qiita1629_mapping.csv"
#p0 <- read_biom2phyloseq(biom.file = biom.file,
  #metadata.file = meta.file,
  #taxonomy.file = NULL)
read_csv2phyloseq  Read Simple OTU Tables into a Phyloseq Object

Description
Read simple OTU tables, mapping and taxonomy files into a phyloseq-class object.

Usage
read_csv2phyloseq(
  otu.file = NULL,
  taxonomy.file = NULL,
  metadata.file = NULL,
  sep = ",",
)

Arguments
otu.file A simple otu_table with `.csv` extension
taxonomy.file A simple taxonomy file with `.csv` extension
metadata.file A simple metadata/mapping file with .csv extension
sep CSV file separator

Details
Simple OTU tables, mapping and taxonomy files will be converted to phyloseq-class.

Value
phyloseq-class object.

Author(s)
Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

Examples
# NOTE: the system.file command reads these example files from the
# microbiome R package. To use your own local files, simply write
# otu.file <- "/path/to/my/file.csv" etc.

#otu.file <-
# system.file("extdata/qiita1629_otu_table.csv",
#   package='microbiome')

tax.file <- system.file("extdata/qiita1629_taxonomy_table.csv",
#   package='microbiome')
read_mothur2phyloseq

# Read Mothur Output into a Phyloseq Object

Description

Read mothur shared and consensus taxonomy files into a phyloseq-class object.

Usage

read_mothur2phyloseq(shared.file, consensus.taxonomy.file, mapping.file = NULL)

Arguments

shared.file A shared file produced by mothur. Identified from the .shared extension
consensus.taxonomy.file Consensus taxonomy file produced by mothur. Identified from with the .taxon-
mapping.file Metadata/mapping file with .csv extension

Details

Mothur shared and consensus taxonomy files will be converted to phyloseq-class.

Value

phyloseq-class object.

Author(s)

Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

Examples

#shared.file <- system.file("extdata/qiita1629_mapping_subset.csv",
#  package='microbiome')

#p0 <- read_csv2phyloseq(
#  otu.file=otu.file,
#  taxonomy.file=tax.file,
#  metadata.file=meta.file)

#otu.file <- system.file("extdata/Baxter_FITs_Microbiome_2016_fit.final.tx.1.subsample.shared",
#  package='microbiome')

#tax.file <- system.file("extdata/Baxter_FITs_Microbiome_2016_fit.final.tx.1.cons.taxonomy",
#  package='microbiome')
# meta.file <- system.file(
#  "extdata/Baxter_FITs_Microbiome_2016_mapping.csv",
#  package='microbiome')

# p0 <- read_mothur2phyloseq(
#  shared.file=otu.file,
#  consensus.taxonomy.file=tax.file,
#  mapping.file=meta.file)

## read_phyloseq

### Import phyloseq Data

**Description**

Read the otu, taxonomy and metadata from various formats.

**Usage**

```r
read_phyloseq(
  otu.file = NULL,
  taxonomy.file = NULL,
  metadata.file = NULL,
  type = c("simple", "mothur", "biom"),
  sep = ","
)
```

**Arguments**

- **otu.file** File containing the OTU table (for mothur this is the file with the .shared extension)
- **taxonomy.file** (for mothur this is typically the consensus taxonomy file with the .taxonomy extension)
- **metadata.file** File containing samples x variables metadata.
- **type** Input data type: 'mothur' or 'simple' or 'biom' type.
- **sep** CSV file separator

**Details**

See help(read_mothur2phyloseq) for details on the Mothur input format; and help(read_biom2phyloseq) for details on the biom format. The simple format refers to the set of CSV files.

**Value**

- **phyloseq-class** object
remove_samples

Author(s)
Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

Examples

```r
# pseq <- read_phyloseq(otu.file,
# taxonomy.file,
# metadata.file,
# type=c('mothur', 'simple', 'biom'))
```

---

remove_samples  Exclude Samples

Description
Filter out selected samples from a phyloseq object.

Usage

```r
remove_samples(samples = NULL, x)
```

Arguments

- **samples** Names of samples to be removed.
- **x** `phyloseq-class` object

Details
This complements the phyloseq function `prune_samples` by providing a way to exclude given groups from a phyloseq object.

Value
Filtered phyloseq object.

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References
To cite the microbiome R package, see citation('microbiome')

See Also
- `phyloseq::prune_samples`, `phyloseq::subset_samples`
**Examples**

```r
data(dietswap)
pseq <- remove_samples(c("Sample-182", "Sample-222"), dietswap)
```

**Description**

Filter out selected taxa from a phyloseq object.

**Usage**

```r
remove_taxa(taxa = NULL, x)
```

**Arguments**

- `taxa`: Names of taxa to be removed.
- `x`: phyloseq-class object

**Details**

This complements the phyloseq function `prune_taxa` by providing a way to exclude given groups from a phyloseq object.

**Value**

Filtered phyloseq object.

**Author(s)**

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

**References**

To cite the microbiome R package, see `citation('microbiome')`

**See Also**

phyloseq::prune_taxa, phyloseq::subset_taxa

**Examples**

```r
data(dietswap)
pseq <- remove_taxa(c("Akkermansia", "Dialister"), dietswap)
```
Description
Community richness index.

Usage
richness(x, index = c("observed", "chao1"), detection = 0)

Arguments
x A species abundance vector, or matrix (taxa/features x samples) with the absolute count data (no relative abundances), or phylloseq-class object
index "observed" or "chao1"
detection Detection threshold. Used for the "observed" index.

Details
By default, returns the richness for multiple detection thresholds defined by the data quantiles. If the detection argument is provided, returns richness with that detection threshold. The "observed" richness corresponds to index="observed", detection=0.

Value
A vector of richness indices

Author(s)
Contact: Leo Lahti <microbiome-admin@googlegroups.com>

See Also
alpha

Examples
data(dietswap)
d <- richness(dietswap, detection=0)
Description

Visualize abundance spread for OTUs

Usage

spreadplot(x, trunc = 0.001/100, alpha = 0.15, width = 0.35)

Arguments

x  phyloseq-class object; or a data.frame with fields "otu" (otu name); "sample" (sample name); and "abundance" (otu abundance in the given sample)
trunc  Truncate abundances lower than this to zero
alpha  Alpha level for point transparency
width  Width for point spread

Value

ggplot2 object

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')

Examples

data(dietswap)
p <- spreadplot(transform(dietswap, "compositional"))
summarize_phyloseq

**Summarize phyloseq object**

**Description**

Prints basic information of data.

**Usage**

`summarize_phyloseq(x)`

**Arguments**

- `x` Input is a `phyloseq-class` object.

**Details**

The `summarize_phyloseq` function will give information on whether data is compositional or not, reads (min. max, median, average), sparsity, presence of singletons and sample variables.

**Value**

Prints basic information of `phyloseq-class` object.

**Author(s)**

Contact: Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

**Examples**

```r
data(dietswap)
summarize_phyloseq(dietswap)
```

---

taxa

**Taxa Names**

**Description**

List the names of taxonomic groups in a phyloseq object.

**Usage**

`taxa(x)`

**Arguments**

- `x` `phyloseq-class` object
Details

A handy shortcut for phyloseq::taxa_names, with a potential to add to add some extra tweaks later.

Value

A vector with taxon names.

Author(s)

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

References

To cite the microbiome R package, see citation('microbiome')

Examples

data(dietswap)
x <- taxa(dietswap)

Usage

otu_tibble(x, column.id = "FeatureID")
tax_tibble(x, column.id = "FeatureID")
sample_tibble(x, column.id = "SampleID")
combine_otu_tax(x, column.id = "FeatureID")

Arguments

x phyloseq-class object.
column.id Provide name for the column which will hold the rownames of slot.

Details

Convert different phyloseq slots into tibbles. otu_tibble gets the otu_table in tibble format. tax_tibble gets the taxa_table in tibble format. combine_otu_tax combines otu_table and taxa_table into one tibble.
timesplit

Value

A tibble

Author(s)

Contact: Sudarshan A. Shetty <sudarshanshetty9@gmail.com>

Examples

```r
library(microbiome)
data("dietswap")
otu_tib <- otu_tibble(dietswap,column.id="FeatureID")
tax_tib <- tax_tibble(dietswap,column.id="FeatureID")
sample_tib <- sample_tibble(dietswap,column.id="SampleID")
otu_tax <- combine_otu_tax(dietswap,column.id = "FeatureID")
head(otu_tax)
```

---

timesplit  Time Split

Description

For each subject, return temporally consecutive sample pairs together with the corresponding time difference.

Usage

```r
timesplit(x)
```

Arguments

- `x`  phylseq object.

Value

data.frame

Author(s)

Leo Lahti <leo.lahti@iki.fi>

Examples

```r
data(atlas1006)
x <- timesplit(subset_samples(atlas1006,
   DNA_extraction_method == 'r' & sex == "male"))
```
time_normalize

Normalize Phyloseq Metadata Time Field

Description
Shift the time field in phyloseq sample_data such that the first time point of each subject is always 0.

Usage
time_normalize(x)

Arguments
x  phyloseq object. The sample_data(x) should contain the following fields: subject, time

Value
Phyloseq object with a normalized time field

Examples
data(peerj32)
pseq <- time_normalize(peerj32$phyloseq)

time_sort

Temporal Sorting Within Subjects

Description
Within each subject, sort samples by time and calculate distance from the baseline point (minimum time).

Usage
time_sort(x)

Arguments
x  A metadata data.frame including the following columns: time, subject, sample, signal. Or a phyloseq object.

Value
A list with sorted metadata (data.frame) for each subject.
Identify Top Entries

Description

Identify top entries in a vector or given field in data frame.

Usage

top(
  x,
  field = NULL,
  n = NULL,
  output = "vector",
  round = NULL,
  na.rm = FALSE,
  include.rank = FALSE
)

Arguments

x  data.frame, matrix, or vector
field  Field or column to check for a data.frame or matrix
n  Number of top entries to show
output  Output format: vector or data.frame
round  Optional rounding
na.rm  Logical. Remove NA before calculating the statistics.
include.rank  Include ranking if the output is data.frame. Logical.

Value

Vector of top counts, named by the corresponding entries
top_taxa

Author(s)
Leo Lahti <leo.lahti@iki.fi>

References
See citation("bibliographica")

Examples
```
data(dietswap)
p <- top(meta(dietswap), "group", 10)
```

---

top_taxa  

Top Taxa

Description
Return n most abundant taxa (based on total abundance over all samples), sorted by abundance

Usage
```
top_taxa(x, n = ntaxa(x))
```

Arguments
- `x`  
  phyloseq object
- `n`  
  Number of top taxa to return (default: all)

Value
Character vector listing the top taxa

Examples
```
data(dietswap)
topx <- top_taxa(dietswap, n=10)
```
Data Transformations for phyloseq Objects

**Description**

Standard transforms for `phyloseq-class`.

**Usage**

```r
transform(
  x,
  transform = "identity",
  target = "OTU",
  shift = 0,
  scale = 1,
  log10 = TRUE,
  reference = 1,
  ...
)
```

**Arguments**

- `x` *phyloseq-class* object
- `transform` Transformation to apply. The options include: 'compositional' (ie relative abundance), 'Z', 'log10', 'log10p', 'hellinger', 'identity', 'clr', 'alr', or any method from the vegan::decostand function.
- `target` Apply the transform for 'sample' or 'OTU'. Does not affect the log transform.
- `shift` A constant indicating how much to shift the baseline abundance (in transform='shift')
- `scale` Scaling constant for the abundance values when transform = "scale".
- `log10` Used only for Z transformation. Apply log10 before Z.
- `reference` Reference feature for the alr transformation.
- `...` arguments to be passed

**Details**

In transformation typ, the 'compositional' abundances are returned as relative abundances in [0, 1] (convert to percentages by multiplying with a factor of 100). The Hellinger transform is square root of the relative abundance but instead given at the scale [0,1]. The log10p transformation refers to log10(1 + x). The log10 transformation is applied as log10(1 + x) if the data contains zeroes. CLR transform applies a pseudocount of min(relative abundance)/2 to exact zero relative abundance entries in OTU table before taking logs.

**Value**

Transformed `phyloseq` object
Examples

data(dietswap)
x <- dietswap

# No transformation
xt <- transform(x, 'identity')

# OTU relative abundances
# xt <- transform(x, 'compositional')

# Z-transform for OTUs
# xt <- transform(x, 'Z', 'OTU')

# Z-transform for samples
# xt <- transform(x, 'Z', 'sample')

# Log10 transform (log10(1+x) if the data contains zeroes)
# xt <- transform(x, 'log10')

# Log10p transform (log10(1+x) always)
# xt <- transform(x, 'log10p')

# CLR transform
# Note that small pseudocount is added if data contains zeroes
xt <- microbiome::transform(x, 'clr')

# ALR transform
# The pseudocount must be specified explicitly
# The reference feature is 1 by default
xt <- microbiome::transform(x, 'alr', shift=1, reference=1)

# Shift the baseline
# xt <- transform(x, 'shift', shift=1)

# Scale
# xt <- transform(x, 'scale', scale=1)

---

ztransform (Z Transformation)

Description

Z transform for matrices

Usage

ztransform(x, which, log10 = TRUE)
**Arguments**

- **x**: a matrix
- **which**: margin
- **log10**: apply log10 transformation before Z

**Details**

Performs centering (to zero) and scaling (to unit variance) across samples for each taxa.

**Value**

Z-transformed matrix

**Author(s)**

Contact: Leo Lahti <microbiome-admin@googlegroups.com>

**References**

See citation('microbiome')

**Examples**

```r
#data(peerj32)
#pseqz <- ztransform(abundances(peerj32$phyloseq))
```
Index

* data
  atlas1006, 11
dietswap, 29
hitchip.taxonomy, 44
peerj32, 58
* early-warning
  potential_univariate, 71
* internal
  chunk_reorder, 18
  estimate_stability, 35
  quiet, 75
  radial_theta, 75
  ztransform, 96
* package
  microbiome-package, 4
* utilities
  abundances, 4
  add_besthit, 5
  add_refseq, 6
  aggregate_rare, 7
  aggregate_taxa, 8
  alpha, 9
  associate, 10
  baseline, 12
  bimodality, 13
  bimodality_sarle, 15
  boxplot_abundance, 16
  boxplot_alpha, 17
  cmat2table, 19
  collapse_replicates, 20
  core, 21
  core_abundance, 22
  core_heatmap, 23
  core_matrix, 24
  core_members, 25
  coverage, 26
  default_colors, 27
  densityplot, 27
  divergence, 30
diversity, 31
dominance, 32
dominant, 34
evenness, 36
find_optima, 37
gktau, 38
group_age, 39
group_bmi, 41
heat, 42
hotplot, 45
inequality, 46
intermediate_stability, 47
is_compositional, 48
log_modulo_skewness, 49
low_abundance, 50
map_levels, 51
merge_taxa2, 52
meta, 53
multimodality, 53
neat, 55
neatsort, 56
overlap, 57
plot_atlas, 59
plot_composition, 60
plot_core, 61
plot_density, 62
plot_frequencies, 63
plot_landscape, 64
plot_regression, 66
plot_taxa_prevalence, 68
plot_tipping, 69
prevalence, 73
psmelt2, 74
rare, 76
rare_abundance, 77
rare_members, 78
rarity, 79
read_biom2phyloseq, 81
read_csv2phyloseq, 82
INDEX

read_mothur2phyloseq, 83
read_phyloseq, 84
readcount, 80
remove_samples, 85
remove_taxa, 86
richness, 87
spreadplot, 88
summarize_phyloseq, 89
taxa, 89
time_sort, 92
timesplit, 91
top, 93
transform, 95
dominant, 34

estimate_richness, 9
estimate_stability, 35

filter_prevalent (core), 21
find_optima, 37
ggplot, 17, 18, 45, 61, 63–65, 68, 69
gktau, 38
group_age, 39
group_bmi, 41

heat, 42
hitchip.taxonomy, 44
hotplot, 45

inequality, 46
intermediate_stability, 47
is_compositional, 48

log_modulo_skewness, 49
low_abundance, 50

map_levels, 51
merge_taxa2, 52
meta, 53
microbiome (microbiome-package), 4
microbiome-package, 4
multimodality, 53

neat, 55
neatsort, 56, 60

ordinate, 56
overlap, 57

peerj32, 58
phyloseq, 12, 24, 51, 61, 73, 95
phyloseq-class, 5, 74, 90
plot_atlas, 59
plot_composition, 60
plot_core, 61
plot_density, 62
plot_frequencies, 63
plot_landscape, 64
plot_regression, 66

ab, (abundances), 4
abundances, 4
add_besthit, 5
add_refseq, 6
aggregate_rare, 7
aggregate_taxa, 8
alpha, 9
associate, 10

atlas1006, 11
baselines, 12
bimodality, 13
bimodality_sarle, 15
boxplot_abundance, 16
boxplot_alpha, 17

collapse_replicates, 20
combine_otu_tax (TibbleUtilities), 90
core, 21
core_abundance, 22
core_heatmap, 23
core_matrix, 24
core_members, 25
coverage, 26
cross_correlate (associate), 10
data.frame, 63
default_colors, 27
densityplot, 27
dietswap, 29
divergence, 30
diversity, 31
dominance, 32

dread_mothur2phyloseq, 83
read_phyloseq, 84
readcount, 80
remove_samples, 85
remove_taxa, 86
richness, 87
spreadplot, 88
summarize_phyloseq, 89
taxa, 89
time_sort, 92
timesplit, 91
top, 93
transform, 95
dominant, 34

estimate_richness, 9
estimate_stability, 35
evenness, 36

filter_prevalent (core), 21
find_optima, 37
ggplot, 17, 18, 45, 61, 63–65, 68, 69
gktau, 38
group_age, 39
group_bmi, 41

heat, 42
hitchip.taxonomy, 44
hotplot, 45

inequality, 46
intermediate_stability, 47
is_compositional, 48

log_modulo_skewness, 49
low_abundance, 50

map_levels, 51
merge_taxa2, 52
meta, 53
microbiome (microbiome-package), 4
microbiome-package, 4
multimodality, 53

neat, 55
neatsort, 56, 60

ordinate, 56
overlap, 57

peerj32, 58
phyloseq, 12, 24, 51, 61, 73, 95
phyloseq-class, 5, 74, 90
plot_atlas, 59
plot_composition, 60
plot_core, 61
plot_density, 62
plot_frequencies, 63
plot_landscape, 64
plot_regression, 66
plot_taxa_prevalence, 68
plot_tipping, 69
potential_analysis, 70
potential_univariate, 71
prevalence, 73
prevalent_taxa (core_members), 25
psmelt2, 74
quiet, 75
radial_theta, 75
rare, 76
rare_abundance, 77
rare_members, 78
rarity, 79
read_biom2phyloseq, 81
read_csv2phyloseq, 82
read_mothur2phyloseq, 83
read_phyloseq, 84
readcount, 80
remove_samples, 85
remove_taxa, 86
richness, 87
sample_data, 53
sample_tibble (TibbleUtilities), 90
spreadplot, 88
summarize_phyloseq, 89
tax_tibble (TibbleUtilities), 90
taxa, 89
TibbleUtilities, 90
time_normalize, 92
time_sort, 92
timesplit, 91
top, 93
top_taxa, 94
transform, 60, 95
vegdist, 55
ztransform, 96