Package 'microbiome'

March 2, 2017

Type Package

Title Tools for microbiome analysis

Encoding UTF-8

Version 0.99.92

Date 2017-02-27

biocViews Microarray, Preprocessing, SystemsBiology, Visualization, Microbiome, Clustering

Description Utilities for microbiome analysis.

License BSD_2_clause + file LICENSE

Depends R (>= 3.0.1), phyloseq

Imports ade4, compositions, dplyr, ggplot2, MASS, moments, plyr, reshape2, tgp, tidyr, vegan, WGCNA

Suggests BiocGenerics, BiocStyle, FD, Hmisc, knitcitations, knitr, magrittr, netresponse, rmarkdown, testthat

URL http://microbiome.github.com

MailingList microbiome <microbiome-devel@googlegroups.com>

BugReports https://github.com/microbiome/microbiome/issues

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Author Leo Lahti [aut, cre],
Tineka Blake [ctb],
Jarkko Salojarvi [ctb]

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

R topics documented:

`microbiome-package` *R package for microbiome studies*

Description

Brief summary of the microbiome package

Details

Package:	microbiome
Type:	Package
Version:	See sessionInfo() or DESCRIPTION file
Date:	2014-2017
License:	FreeBSD
LazyLoad:	yes

R package for microbiome studies

Author(s)

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

References

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

Examples

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

abundances *Abundance matrix from phyloseq object*

Description

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

Usage

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

Arguments

x	phyloseq-class object
transform	Transformation to apply. The options include: 'compositional' (ie relative abundance), 'Z', 'log10', 'hellinger', 'identity', 'clr', 'ilr', or any method from the vegan::decostand function.

Value

Abundance matrix.

Author(s)

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

References

See citation('microbiome')

Examples

```
data(dietswap)
abundances(dietswap)
abundances(dietswap, transform = "identity")
abundances(dietswap, transform = "compositional")
abundances(dietswap, transform = "clr")
abundances(dietswap, transform = "Z")
abundances(dietswap, transform = "log10")
```

aggregate_taxa *Summarize Taxa*

Description

Summarize phyloseq data into a higher phylogenetic level.

Usage

```
aggregate_taxa(pseq, level)
```

Arguments

pseq	phyloseq-class object
level	Summarization level (from rank_names(pseq))

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")
```

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', 'spearman', or 'bicor' for continuous; categorical for discrete)
p.adj.threshold	q-value threshold to include features
cth	correlation threshold to include features
order	order the results
n.signif	mininum 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

As the method=categorical (discrete) association measure for nominal (no order for levels) variables we use Goodman and Kruskal tau based on r-bloggers.com/measuring-associations-between-non-numeric-variables/

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]
cc <- associate(d1, d2, method = "pearson")
```

---

atlas1006 *HITChip Atlas with 1006 Western Adults*

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) http://www.nature.com/ncomms/2014/140708/ncomms5344/full/ncomms5344.html.

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.

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')

Bagged.RDA *Bagged RDA*

Description

Bootstrap solutions that follows the Jack-knife estimation of PLS by Martens and Martens, 2000. Solves rotational invariance of latent space by orthogonal procrustes rotations.

Usage

```
Bagged.RDA(X, Y, boot = 1000)
```

Arguments

X	a matrix, samples on columns, variables (bacteria) on rows.
Y	vector with names(Y)=rownames(X), for example
boot	Number of bootstrap iterations

Value

List with elements:

- loadingsbagged loadings
- scoresbagged scores
- significancesignificances of X variables

Author(s)

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

References

See citation("microbiome")

Examples

```
 ## Not run:
   data(peerj32)
   x <- as.matrix(peerj32$microbes)[1:20, 1:6]
   y <- rnorm(nrow(x))
   names(y) <- rownames(x)
   res <- Bagged.RDA(x, y , boot = 5)

## End(Not run)
```

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(atlas1006)
a <- baseline(atlas1006)
```

`bimodality` *Bimodality Analysis*

Description

A wrapper to calculate bimodality scores.

Usage

```
bimodality(x, method = "potential_analysis", detection = 1, bw.adjust = 1,
  bs.iter = 100, detection.limit = 1, verbose = TRUE)
```

Arguments

`x`	A vector, matrix, or a phyloseq object
`method`	bimodality quantification method ('potential_analysis' or one of the methods in bimodality_sarle)
`detection`	Mode detection
`bw.adjust`	Bandwidth adjustment
`bs.iter`	Bootstrap iterations
`detection.limit`	minimum accepted density for a maximum; as a multiple of kernel height
`verbose`	Verbose

Details

- Sarle.finite.sampleCoefficient of bimodality for finite sample. See SAS 2012.
- Sarle.asymptoticCoefficient of bimodality, used and described in Shade et al. (2014) and Ellison AM (1987).
- potential_analysisRepeats 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.

Value

A list with following elements:

- scoreFraction of bootstrap samples where multiple modes are observed
- nmodesThe most frequently observed number of modes in bootrstrap sampling results
- resultsFull 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.
- Lahti et al. (2014). Tipping elements of the human intestinal ecosystem. *Nature Communications* 5:4344.
- Shade et al. mBio 5(4):e01371-14, 2014.
- AM Ellison, Am. J. Bot 74:1280-8, 1987.
- SAS Institute Inc. (2012). SAS/STAT 12.1 user's guide. Cary, NC.

See Also

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

Examples

```
 bimodality(c(rnorm(100, mean = 0), rnorm(100, mean = 5)))
#
# See also the classical DIP test:
# Dip 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.
```

`bimodality_sarle` *Sarle's Bimodality Coefficient*

Description

Sarle's bimodality coefficient.

Usage

```
bimodality_sarle(x, bs.iter = 1, na.rm = TRUE,
  type = "Sarle.finite.sample")
```

Arguments

`x`	Data vector for which bimodality will be quantified
`bs.iter`	Bootstrap iterations
`na.rm`	Remove NAs
`type`	Score type ("Sarle.finite.sample" or "Sarle.asymptotic")

Details

The coefficient lies in (0, 1).

The 'Sarle.asymptotic' version is defined as

$$b = (g^2 + 1)/k$$

. 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

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>

References

- Shade et al. mBio 5(4):e01371-14, 2014.
- Ellison AM (1987) Am J Botany 74(8):1280-1288.
- SAS Institute Inc. (2012). SAS/STAT 12.1 user's guide. Cary, NC.
- 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

```
bimodality_sarle(rnorm(100), type = "Sarle.finite.sample")
```

boxplot_abundance *Abundance Boxplot*

Description

Plot phyloseq abundances.

Usage

```
boxplot_abundance(pseq, x, y, line = NULL, color = NULL, log10 = FALSE,
  violin = FALSE, na.rm = FALSE, show.points = TRUE)
```

Arguments

pseq	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
color	The variable to map on colors
log10	show y axis on log scale
violin	Use violin version of the boxplot
na.rm	Remove NAs
show.points	Include data points in the figure

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", color = "gender")
```

cmat2table *Convert Cross Correlation Results To 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 <- cmat2table(cc)
```

core *Core Microbiota*

Description

Filter the phyloseq object to include only prevalent taxa.

Usage

```
core(x, detection, prevalence, method = "standard", Nsample = NULL,
  bs.iter = 1000, I.max = NULL)
```

Arguments

x	phyloseq-class object
detection	Detection threshold (non-negative real)
prevalence	Prevalence threshold (in [0, 100])
method	Either "standard" or "bootstrap". The standard methods selects the taxa that exceed the given detection and prevalence threshold. The bootstrap method is more robust an described in Salonen et al. (2012). Note that the results may depend on the random seed unless a sufficiently large bootstrap sample size is used.
Nsample	Only needed for method "bootstrap". Bootstrap sample size, default is the same size as data.
bs.iter	Only needed for method "bootstrap". Bootstrap iterations.
I.max	Only needed for method "bootstrap". Upper limit for intensity threshold. Later addition. Set to NULL (default) to replicate Salonen et al.
...	Arguments to pass.

Value

Filtered phyloseq object including only prevalent taxa

Author(s)

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

References

The core microbiota bootstrap method implemented with this function: 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')

Examples

```
data(peerj32)
core(peerj32$phyloseq, 200, 20)
```

---

core_bootstrap *Bootstrap Analysis of the Core Microbiota*

Description

Bootstrap analysis of the core microbiota.

Usage

```
core_bootstrap(x, Nsample = NULL, prevalence = 2, bs.iter = 1000,
  detection = 1.8, I.max = NULL)
```

Arguments

x	OTUxSample data matrix
Nsample	bootstrap sample size, default is the same size as data
prevalence	Lower limit for number of samples where microbe needs to exceed the intensity threshold for a 'present' call.
bs.iter	bootstrap iterations
detection	Lower limit for intensity threshold
I.max	Upper limit for intensity threshold. Later addition. set to NULL (default) to replicate Salonen et al.

Value

data frame with microbes and their frequency of presence in the core

Author(s)

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

References

The core microbiota bootstrap method implemented with this function: 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 this R package, see citation("microbiome")

Examples

```
data(peerj32)
# In practice, use bs.iter = 1000 or more
bs <- core_bootstrap(peerj32$phyloseq, bs.iter = 5)
```

---

core_heatmap *Core Heatmap*

Description

Core heatmap.

Usage

```
core_heatmap(data, detections = 20, colours = gray(seq(0, 1, length = 5)),
  min.prevalence = NULL, taxa.order = NULL)
```

Arguments

data	OTU matrix
detections	A vector or a scalar indicating the number of intervals in (0, log10(max(data))). The detections are calculated for relative abundancies.
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.
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 *Core Matrix*

Description

Creates the core matrix.

Usage

```
core_matrix(x, prevalences = seq(5, 100, 5), detections = NULL)
```

Arguments

x	phyloseq object or a taxa x samples abundance matrix
prevalences	a vector of prevalence percentages in [0,100]
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

```
library(microbiome)
data(peerj32)
core <- core_matrix(peerj32$phyloseq)
```

---

core_members *Core Taxa*

Description

Determine members of the core microbiota with given abundance and prevalences

Usage

```
core_members(x, detection = 1, prevalence = 95, method = "standard",
  Nsample = NULL, bs.iter = 1000, I.max = NULL)
```

Arguments

x	phyloseq-class object
detection	Detection threshold (non-negative real)
prevalence	Prevalence threshold (in [0, 100])
method	Either "standard" or "bootstrap". The standard methods selects the taxa that exceed the given detection and prevalence threshold. The bootstrap method is more robust an described in Salonen et al. (2012). Note that the results may depend on the random seed unless a sufficiently large bootstrap sample size is used.
Nsample	Only needed for method "bootstrap". Bootstrap sample size, default is the same size as data.
bs.iter	Only needed for method "bootstrap". Bootstrap iterations.
I.max	Only needed for method "bootstrap". Upper limit for intensity threshold. Later addition. Set to NULL (default) to replicate Salonen et al.

Details

For phyloseq object, lists taxa that are more prevalent with the given detection. 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)
a <- core_members(dietswap, 1, 95)
```

---

densityplot *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)
```

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

Value

ggplot2 object

Author(s)

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

References

See citation('microbiome')

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 http://www.nature.com/ncomms/2015/150428/ncomms7342/full/ncomms7342.html.

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.

Author(s)

Leo Lahti <microbiome-admin@googlegroups.com>

References

O'Keefe et al. Nature Communications 6:6342, 2015. http://www.nature.com/ncomms/2015/150428/ncomms7342/full/ncomms7342.html To cite the microbiome R package, see citation('microbiome')

diversity *Estimate Diversity*

Description

Diversity estimation. Augments the estimate_richness function of the phyloseq package.

Usage

```
diversity(x, detection = 0, split = TRUE, measures = NULL)
```

Arguments

x	phyloseq-class object
detection	detection for observing taxa (absence / presence). Used to determine the richness (Observed diversity) above this abundance threshold. Zero by default.
split	(Optional). Logical. Should a separate set of richness estimates be performed for each sample? Or alternatively, pool all samples and estimate richness of the entire set.
measures	(Optional). Default is 'NULL', meaning that all available alpha-diversity measures will be included. Alternatively, you can specify one or more measures as a character vector of measure names. Values must be among those supported in the phyloseq::estimate_richness function. These include 'c("Observed", "Chao1", "ACE", "Shannon", "Simpson", "InvSimpson", "Fisher")'. In addition, the measure "Evenness" is provided (Pielou's index).

Value

A data.frame of samples x diversity indicators; except when split=FALSE, a vector of indices is returned.

Author(s)

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

References

See citation('microbiome')

Examples

```
data(dietswap)
d <- diversity(dietswap)
```

`estimate_stability` *Estimate Stability*

Description

Quantify intermediate stability with respect to a given reference point.

Usage

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

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

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

```
 ## Not run:
  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")

## End(Not run)
```

`find_optima` *Find Optima*

Description

Detect optima, excluding local optima below detection.

Usage

```
find_optima(f, detection = 0, bw = 1, detection.limit = 1)
```

Arguments

`f`	density
`detection`	detection for peaks
`bw`	bandwidth
`detection.limit`	Minimun accepted density for a maximum; as a multiple of kernel height

Value

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

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

See citation('microbiome')

Examples

```
find_optima(rnorm(100), bw = 1)
```

get_ordination *Get Ordination*

Description

Ordinate phyloseq data and merge it with sample metadata

Usage

```
get_ordination(x, method = "NMDS", distance = "bray")
```

Arguments

x	phyloseq-class object or a data matrix (features x samples; eg. HITChip taxa vs. samples)
method	Ordination method, see phyloseq::plot_ordination
distance	Ordination distance, see phyloseq::plot_ordination

Details

This is a wrapper for phyloseq ordination functions, providing smooth access to ordinated data.frame with full info on the projection and metadata necessary for further visualizations.

Value

data.frame with ordination coordinates and metadata

Author(s)

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

References

See citation('microbiome')

See Also

phyloseq::plot_ordination

Examples

```
data(dietswap)
fc <- get_ordination(dietswap)
```

GKtau *GKtau*

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

Examples

```
data(peerj32)
v1 <- unlist(peerj32$microbes[,1])
v2 <- unlist(peerj32$lipids[,1])
tc <- GKtau(v1, v2)
```

`group_diversity` *Diversity within a Sample Group*

Description

Quantify microbiota heterogeneity within a given sample set.

Usage

```
group_diversity(x, method = "anticorrelation")
```

Arguments

`x`	phyloseq object
`method`	dissimilarity method ("anticorrelation" or any method available via the vetan::vegdist function)

Details

Microbiota heterogeneity within a given sample set can be quantified by the average sample dissimilarity or beta diversity. Taking average over all pairwise dissimilarities is sensitive to sample size and heavily biased as the similarity values are not independent. To reduce this bias, the dissimilarity of each sample against the group mean is calculated. This generates one value per sample. These can be compared between groups in order to compare differences in group homogeneity.

Note that this measure is still affected by sample size. Subsampling or bootstrapping can be applied to equalize sample sizes between comparisons.

The anticorrelation mode is a simple indicator that returns average spearman correlation between samples of the input data and the overall group-wise average. The inverse of this measure (ie cor instead of 1-cor as in here) was used in Salonen et al. (2014) to quantify group homogeneity.

Value

Vector with dissimilarities; one for each sample, quantifying the dissimilarity of the sample from the group-level mean.

Author(s)

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

References

The inter- and intra-individual homogeneity measures used in Salonen et al. ISME J. 8:2218-30, 2014 were obtained as 1 - beta where beta is the group diversity as quantified by the anticorrelation method.

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

See Also

the vegdist function from the **vegan** package provides many standard beta diversity measures

Examples

```
# Example data
data(peerj32)
# Assess beta diversity among the African samples
# in a diet swap study
b <- group_diversity(subset_samples(dietswap, group == "AFR"))
```

---

heat *Association Heatmap*

Description

Visualizes n x m association table as heatmap.

Usage

```
heat(df, Xvar, Yvar, fill, star, p.adj.threshold = 1,
  association.threshold = 0, step = 0.2, colours = c("darkblue", "blue",
  "white", "red", "darkred"), limits = NULL, legend.text = "",
  order.rows = TRUE, order.cols = TRUE, text.size = 10,
  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
order.cols	Order columns to enhance visualization interpretability
text.size	Adjust text size
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")
```

---

hitchip.taxonomy *HITChip Taxonomy*

Description

HITChip taxonomy table.

Usage

```
data(hitchip.taxonomy)
```

Format

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

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

```
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	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
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

```
data(atlas1006)
pseq <- atlas1006
pseq <- subset_samples(pseq, DNA_extraction_method == "r")
# Bimodality is often best visible at log10 relative abundances
pseq <- transform(transform(pseq, "compositional"), "log10")
p <- hotplot(pseq, "Dialister", tipping.point = .3)
```

`intermediate_stability` *Intermediate Stability*

Description

Quantify intermediate stability with respect to a given reference point.

Usage

```
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

Author(s)

Leo Lahti <leo.lahti@iki.fi>

Examples

```
## Not run:
  library(microbiome)
  data(atlas1006)
  res <- intermediate_stability(x, reference.point = NULL)
  s <- sapply(res, function (x) {x$stability})

## End(Not run)
```

`is.phyloseq` *Identify Phyloseq Objects*

Description

Identifies whether a given object is from the phyloseq class

Usage

```
is.phyloseq(x)
```

Arguments

`x`	object to test

Value

Logical

Examples

```
library(microbiome)
data(dietswap)
is.phyloseq(dietswap)
```

`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 codetaxonomyTable-class , see the **phyloseq** package.

Value

mappings

Author(s)

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

References

See citation('microbiome')

Examples

```
tax.table <- get_hitchip_taxonomy('HITChip', 'filtered')
map_levels('Akkermansia', 'L2', 'L1', tax.table)
```

---

meta *Retrieve Phyloseq Metadata as 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

```
meta(x)
```

Arguments

x	a phyloseq object

Author(s)

Leo Lahti <leo.lahti@iki.fi>

Examples

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

`multimodality_score` *Multimodality Score*

Description

Multimodality score based on bootstrapped potential analysis.

Usage

```
multimodality_score(x, detection = 1, bw.adjust = 1, bs.iter = 100,
  detection.limit = 1, verbose = TRUE)
```

Arguments

`x`	A vector, or data matrix (variables x samples)
`detection`	Mode detection
`bw.adjust`	Bandwidth adjustment
`bs.iter`	Bootstrap iterations
`detection.limit`	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:

- scoreFraction of bootstrap samples with multiple observed modes
- nmodesThe most frequently observed number of modes in bootstrap
- resultsFull 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.
- Lahti et al. (2014). Tipping elements of the human intestinal ecosystem. *Nature Communications* 5:4344.

Examples

```
data(peerj32)
s <- multimodality_score(
        t(peerj32$microbes[, c("Akkermansia", "Dialister")]))
```

`neat` *Neatmap Sorting*

Description

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

Usage

```
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)
x <- peerj32$microbes
xo <- 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

```
## Not run:
  data(peerj32)
  pseq <- peerj32$phyloseq
  sort.otu <- neatsort(pseq, target = "species")
  sort.rows <- neatsort(abundances(pseq), target = "rows")

## End(Not run)
```

---

peerj32 *Probiotics Intervention Data*

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, gender, subjectID, sampleID and treatment group (probiotic LGG / Placebo)
- phyloseq The microbiome data set converted into a phyloseq-class object.

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)
plot_atlas(atlas1006, "DNA_extraction_method", "diversity")
plot_atlas(atlas1006, "DNA_extraction_method", "Bifidobacterium")
```

plot_composition *Taxonomic Composition Plot*

Description

Plot taxon abundance for samples.

Usage

```
plot_composition(x, taxonomic.level = "OTU", sample.sort = NULL,
  otu.sort = NULL, x.label = "sample", plot.type = "barplot",
  verbose = FALSE, transform = NULL, mar = c(5, 12, 1, 1),
  average_by = NULL, ...)
```

Arguments

x	phyloseq-class object
taxonomic.level	Merge the OTUs (for phyloseq object) into a higher taxonomic level. This has to be one from colnames(tax_table(x)).
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 • 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 metadata, 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 sample_variables(x).
plot.type	Plot type: 'barplot' or 'heatmap'
verbose	verbose
transform	Data transform to be used in plotting (but not in sample/taxon ordering). The options are 'Z-OTU', 'Z-Sample', 'log10' and 'relative.abundance'. See the transform function.
mar	Figure margins
average_by	Average the samples by the average_by variable
...	Arguments to be passed (for neatsort function)

Value

A ggplot plot object.

Examples

```
## Not run:
  # Example data
    library(microbiome)
    data("dietswap")
    pseq <- subset_samples(dietswap, group == "DI" & nationality == "AFR")
    plot_composition(pseq, taxonomic.level = "Phylum")

## End(Not run)
```

`plot_core` *Visualize OTU Core*

Description

Core visualization (2D).

Usage

```
plot_core(x, prevalences = seq(5, 100, 5), detections = 20,
  plot.type = "lineplot", colours = gray(seq(0, 1, length = 5)),
  min.prevalence = NULL, taxa.order = NULL, horizontal = FALSE)
```

Arguments

`x`	A phyloseq object or a core matrix
`prevalences`	a vector of prevalence percentages in [0,100]
`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.
`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(atlas1006)
pseq <- atlas1006
p <- plot_core(pseq, prevalences = seq(10, 100, 10),
                     detections = c(0, 10^(0:4)))
```

`plot_density` *Plot Density*

Description

Plot abundance density across samples for a given taxon.

Usage

```
plot_density(x, variable = NULL, log10 = FALSE, adjust = 1,
  kernel = "gaussian", trim = FALSE, na.rm = FALSE, fill = "gray",
  tipping.point = NULL, xlim = NULL)
```

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

```
p <- plot_density(x, variable = "Dialister")
```

plot_diversity *Plot Diversity*

Description

Plot alpha diversity. This function estimates a number of alpha-diversity metrics using the estimate_richness function, and returns a ggplot object. The plot generated by this function will include every sample in physeq, but they can be further grouped on the horizontal axis through the argument to x, and shaded according to the argument to color (see below). You must use untrimmed, non-normalized count data for meaningful results.

Usage

```
plot_diversity(x, variable = "group", measures = "Shannon", nrow = 1,
  scales = "free_y", detection = 0, indicate.subjects = FALSE,
  na.rm = FALSE)
```

Arguments

x	phyloseq-class object
variable	A variable to map to the horizontal axis. The vertical axis will be mapped to the alpha diversity index/estimate and have units of total taxa, and/or index value (dimensionless). This parameter (x) is a character string indicating a in the dataset (nsamples(x)).
measures	Default is NULL. In this case all available alpha-diversity measures will be included. Alternatively, you can specify one or more measures as a character vector. Values must be among those supported: c("Observed", "Chao1", "ACE", "Shannon", "Simpson'
nrow	Number of rows for plot faceting.
scales	scales for the plot
detection	Detection threshold for the diversity measure 'Observed' (ie. species richness). See diversity
indicate.subjects	Indicate subjects by lines. The sample_data(x) must have 'subject' field.
na.rm	Remove samples with missing metadata (NA)

Details

If subject is among the metadata variables, the matched subjects across groups are indicated by lines.

Value

A ggplot plot object summarizing the richness estimates, and their standard error.

See Also

estimate_richness diversity plot_richness estimateR diversity

Examples

```
p <- plot_diversity(x, variable = "bmi_group", "Shannon")
```

---

plot_frequencies *Plot Frequencies*

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

Value

A list with two elements:

- dataTable with the indicated frequencies.
- plotggplot plot object.

Examples

```
data(dietswap)
p <- plot_frequencies(sample_data(dietswap), "group", "sex")
```

plot_landscape *Landscape Plot*

Description

Plot abundance landscape ie. sample density in 2D projection landscape

Usage

```
plot_landscape(x, method = "NMDS", distance = "bray", col = NULL,
  main = NULL, x.ticks = 10, rounding = 0, add.points = TRUE,
  adjust = 1, size = 1, legend = FALSE)
```

Arguments

x	phyloseq-class object or a data matrix (features x samples; eg. HITChip taxa vs. samples)
method	Ordination method, see phyloseq::plot_ordination
distance	Ordination distance, see phyloseq::plot_ordination
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

Details

For consistent results, set random seet (set.seed) before function call

Value

A ggplot plot object.

Examples

```
## Not run:
  data(dietswap)
  p <- plot_landscape(dietswap)

## End(Not run)
```

plot_matrix *Matrix Heatmap*

Description

Fast investigation of matrix objects; standard visualization choices made automatically.

Usage

```
plot_matrix(mat, type = "twoway", midpoint = 0, palette = NULL,
  colors = NULL, col.breaks = NULL, interval = 0.1, plot_axes = "both",
  row.tick = 1, col.tick = 1, cex.xlab = 0.9, cex.ylab = 0.9,
  xlab = NULL, ylab = NULL, limit.trunc = 0, cap = NULL, mar = c(5, 4,
  4, 2), ...)
```

Arguments

mat	matrix
type	String. Specifies visualization type. Options: 'oneway' (color scale ranges from white to dark red; the color can be changed if needed); 'twoway' (color scale ranges from dark blue through white to dark red; colors can be changed if needed)
midpoint	middle point for the color plot: smaller values are shown with blue, larger are shown with red in type = 'twoway'
palette	Optional. Color palette.
colors	Optional. Colors.
col.breaks	breakpoints for the color palette
interval	interval for palette color switches
plot_axes	String. Indicates whether to plot x-axis ('x'), y-axis ('y'), or both ('both').
row.tick	interval for plotting row axis texts
col.tick	interval for plotting column axis texts
cex.xlab	use this to specify distinct font size for the x axis
cex.ylab	use this to specify distinct font size for the y axis
xlab	optional x axis labels
ylab	optional y axis labels
limit.trunc	color scale rounding
cap	Color scale end point
mar	image margins
...	optional parameters to be passed to function 'image', see help(image) for further details

Value

A list with the color palette (colors), color breakpoints (breaks), and palette function (palette.function)

Author(s)

Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation('microbiome')

Examples

```
mat <- rbind(c(1,2,3,4,5), c(1, 3, 1), c(4,2,2))
res <- plot_matrix(mat, 'twoway', midpoint = 3)
```

---

plot_potential *Plot Potential*

Description

Visualization of the potential function.

Usage

```
plot_potential(res, cutoff = 0.5, plot.contours = TRUE, binwidth = 0.2,
  bins = NULL)
```

Arguments

res	output from potential_slidingaverage function
cutoff	parameter determining the upper limit of potential for visualizations
plot.contours	Plot contour lines.
binwidth	binwidth for contour plot
bins	bins for contour plot. Overrides binwidth if given

Details

Applied on the output of the potential_slidingaverage function.

Value

A ggplot2 visualization of the potential landscape.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Lahti et al. Tipping elements of the human intestinal ecosystem. *Nature Communications* 5:4344, 2014.

Examples

```
X <- c(rnorm(1000, mean = 0), rnorm(1000, mean = -2),
     rnorm(1000, mean = 2))
param <- seq(0,5,length=3000);
res <- potential_slidingaverage(X, param);
plot_potential(res$res, cutoff = 0.5)
```

`plot_rda_bagged` *Plot RDA*

Description

rda_bagged output visualization.

Usage

```
plot_rda_bagged(x, which.bac = 1:nrow(x$loadings), ptype = "spider",
  comp = 1:2, cex.bac = 0.5, plot.names = T,
  group.cols = as.numeric(unique(Y)), ...)
```

Arguments

`x`	Output from rda_bagged
`which.bac`	TBA
`ptype`	Plot type. "spider" or "hull"
`comp`	TBA
`cex.bac`	Plot size.
`plot.names`	Plot names
`group.cols`	Group colors.
`...`	Other arguments to be passed

Value

TBA

Author(s)

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

References

See citation("microbiome")

Examples

```
## Not run:
  library(microbiome)
  data(peerj32)
  x <- t(peerj32$microbes)
  y <- factor(peerj32$meta$time); names(y) <- rownames(peerj32$meta)
  res <- rda_bagged(x, y, detection=0.05, bs.iter=100)
  plot_rda_bagged(res)

## End(Not run)
```

---

plot_regression *Visually Weighted Regression Plot*

Description

Draw regression curve with smoothed error bars with Visually-Weighted Regression by Solomon M. Hsiang; see http://www.fight-entropy.com/2012/07/visually-weighted-regression.html The R is modified from Felix Schonbrodt's original code at http://www.nicebread.de/visually-weighted-watercolor-plots-new-variants-please-vote

Usage

```
plot_regression(formula, data, B = 1000, shade = TRUE, shade.alpha = 0.1,
  spag = FALSE, mweight = TRUE, show.lm = FALSE, show.median = TRUE,
  median.col = "white", show.CI = FALSE, method = loess, bw = FALSE,
  slices = 200, palette = colorRampPalette(c("#FFEDA0", "#DD0000"), bias =
  2)(20), ylim = NULL, quantize = "continuous", show.points = TRUE, ...)
```

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	should the median smoother be visually weighted?
show.lm	should the linear regresison line be plotted?
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
bw	define a default b/w-palette (TRUE/FALSE)
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
palette	provide a custom color palette for the watercolors
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.
...	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 Felix Schonbrodt. Modified by Leo Lahti <microbiome-admin@googlegroups.com>

References

See citation("microbiome")

Examples

```
  ## Not run:
    data(atlas1006)
    p <- plot_regression(diversity ~ age, sample_data(atlas1006))

## End(Not run)
```

---

potential_analysis *Bootstrapped 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, detection, bw.adjust = 1, bs.iter = 100,
  detection.limit = 1)
```

Arguments

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

Value

List with following elements:

- modesNumber of modes for the input data vector (the most frequent number of modes from bootstrap)
- modesminima: Average of potential minima across the bootstrap samples (for the most frequent number of modes)
- modesmaxima: Average of potential maxima across the bootstrap samples (for the most frequent number of modes)
- modesunimodality.support Fraction of bootstrap samples exhibiting unimodality

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.
- Lahti et al. (2014). Tipping elements of the human intestinal ecosystem. *Nature Communications* 5:4344.

---

potential_slidingaverage

Moving Average Potential

Description

This function reconstructs a potential derived from data along a gradient of a given parameter.

Usage

```
potential_slidingaverage(X, param = NULL, bw = "nrd", bw.adjust = 1,
  detection = 0.1, std = 1, grid.size = 50, plot.cutoff = 0.5,
  plot.contours = TRUE, binwidth = 0.2, bins = NULL)
```

Arguments

X	a vector of the X observations of the state variable of interest
param	parameter values corresponding to the observations in X
bw	Bandwidth for smoothing kernels. Automatically determined by default.
bw.adjust	Bandwidth adjustment constant
detection	Threshold for local optima to be discarded.
std	Standard deviation.
grid.size	number of evaluation points; number of steps between min and max potential; also used as kernel window size
plot.cutoff	cuttoff for potential minima and maxima in visualization
plot.contours	Plot contours on the landscape visualization
binwidth	binwidth for contour plot
bins	bins for contour plot. Overrides binwidth if given

Value

A list with the following elements:

- parsvalues of the covariate parameter as matrix
- xisvalues of the x as matrix
- potssmoothed potentials
- minsminima in the densities (-potentials; neglecting local optima)
- maxsmaxima in densities (-potentials; neglecting local optima)
- plotan object that displays the potential estimated in 2D

Author(s)

Leo Lahti, adapted from original Matlab code by Egbert van Nes.

References

- Hirota, M., Holmgren, M., van Nes, E.H. & Scheffer, M. (2011). Global resilience of tropical forest and savanna to critical transitions. *Science*, 334, 232-235.
- Lahti et al. (2014). Tipping elements of the human intestinal ecosystem. *Nature Communications* 5:4344.

See Also

potential_univariate

Examples

```
X <- c(rnorm(1000, mean = 0),
         rnorm(1000, mean = -2),
         rnorm(1000, mean = 2));
    param = seq(0,5,length=3000);
  res <- potential_slidingaverage(X, param)
```

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, detection = 1, bw.adjust = 1, density.smoothing = 0,
  detection.limit = 1)
```

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.
detection	maximum detection as fraction of density kernel height dnorm(0, sd = bandwidth)/N
bw.adjust	The real bandwidth will be bw.adjust*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.
detection.limit	minimum accepted density for a maximum; as a multiple of kernel height

Value

potential_univariate returns a list with the following elements:

- xithe grid of points on which the potential is estimated
- potThe estimated potential: -log(f)*std^2/2, where f is the density.
- densityDensity estimate corresponding to the potential.
- min.indsindices of the grid points at which the density has minimum values; (-potentials; neglecting local optima)
- max.indsindices the grid points at which the density has maximum values; (-potentials; neglecting local optima)
- bwbandwidth of kernel used

- min.pointsgrid point values at which the density has minimum values; (-potentials; neglecting local optima)
- max.pointsgrid 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.
- Lahti et al. (2014). Tipping elements of the human intestinal ecosystem. *Nature Communications* 5:4344.

See Also

potential_slidingaverage

Examples

```
## Not run: res <- potential_univariate(x)
```

---

prevalence *Prevalence for Phyloseq OTUs*

Description

Simple prevalence measure.

Usage

```
prevalence(x, detection = 0, sort = FALSE, count = FALSE)
```

Arguments

x	A vector, data matrix or phyloseq object
detection	Detection threshold for absence/presence.
sort	Sort the groups by prevalence
count	Logical. Indicate prevalence as fraction of samples (in percentage [0, 100]; default); or in absolute counts indicating the number of samples where the OTU is detected above the given abundance threshold.

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)
## With matrix
prevalence(peerj32$data$microbes, detection = 200, sort = TRUE)
## With phyloseq
prevalence(peerj32$phyloseq, detection = 200, sort = TRUE)
prevalence(peerj32$phyloseq, detection = 200, sort = TRUE, count = TRUE)
```

---

pseq_metadata *Retrieve Phyloseq Metadata as 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

```
pseq_metadata(x)
```

Arguments

x	a phyloseq object

Author(s)

Leo Lahti <leo.lahti@iki.fi>

Examples

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

`rare` *Select Rare Taxa*

Description

Filter the phyloseq object to include only rare taxa.

Usage

```
rare(x, detection, prevalence)
```

Arguments

`x`	`phyloseq-class` object
`detection`	Detection threshold (non-negative real)
`prevalence`	Prevalence threshold (in [0, 100])

Value

Filtered phyloseq object including only rare taxa

Author(s)

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

References

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

Examples

```
data(peerj32)
rare(peerj32$phyloseq, 200, 20)
```

rda_bagged *Bagged RDA*

Description

Bagged (or Bootstrap Aggregated) RDA feature selection

Usage

```
rda_bagged(x, y, bs.iter = 1000, verbose = T)
```

Arguments

x	a matrix, samples on columns, variables (bacteria) on rows. Or a phyloseq-class object
y	vector or factor with names(y)=rownames(X). Or name of phyloseq sample data variable name (one of sample_variables(x)).
bs.iter	Number of bootstrap iterations
verbose	verbose

Details

Bootstrap aggregation (bagging) is expected to improve the stability of the results. Aggregating results over several modeling runs with different boostrap samples of the data are averaged to produce the final summary.

Value

List with items:

- loadingsbagged loadings
- significancesignificances of X variables
- scoresbagged scores
- group.centersgroup centers on latent space
- bootstrappedbootstrapped loadings
- datadata set with non-significant components dropped out

Author(s)

Jarkko Salojarvi <microbiome-admin@googlegroups.com>

References

See citation("microbiome")

See Also

vegan::rda and phyloseq::ordinate

Examples

```
## Not run:
  library(microbiome)

  # Example with abundance matrix
  data(peerj32)
  phy <- peerj32$phyloseq
  x <- abundances(phy)
  y <- factor(sample_data(phy)$gender);
  names(y) <- rownames(sample_data(phy))
  res <- rda_bagged(x, y, bs.iter=20)
  plot_rda_bagged(res, y)

  # Example with phyloseq object
  res <- rda_bagged(phy, "gender", bs.iter=20)
  plot_rda_bagged(res, y)


## End(Not run)
```

---

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)
taxa(dietswap)
```

`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(atlas1006)
atlas1006b <- time_normalize(atlas1006)
```

`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.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

See citation("microbiome")

Examples

```
## Not run: time_sort(x)
```

---

tipplot *Variation Line Plot*

Description

Plot variation in taxon abundance for many subjects.

Usage

```
tipplot(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 <- atlas1006
pseq <- subset_samples(pseq, DNA_extraction_method == "r")
pseq <- transform(pseq, "compositional")
p <- tipplot(pseq, "Dialister", tipping.point = 1)
```

---

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)
```

`transform` *Data Transformations for phyloseq Objects*

Description

Standard transforms for phyloseq-class.

Usage

```
transform(x, transform = "identity", target = "OTU")
```

Arguments

x	phyloseq-class object
transform	Transformation to apply. The options include: 'compositional' (ie relative abundance), 'Z', 'log10', 'hellinger', 'identity', 'clr', 'ilr', or any method from the vegan::decostand function.
target	Apply the transform for 'sample' or 'OTU'. Does not affect the log transform.

Details

The relative abunance are returned as percentages in [0, 100]. The Hellinger transform is square root of the relative abundance but instead given at the scale [0,1].

Value

Transformed phyloseq object

Examples

```
## Not run:

  # OTU relative abundances
  xt <- transform(x, "relative.abundance", "OTU")

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

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

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


## End(Not run)
```

validate *Validate Phyloseq*

Description

Validate phyloseq object.

Usage

```
validate(x)
```

Arguments

x	phyloseq object

Details

Checks that the abundances and sample_data have exactly same samples.

Value

A validated and polished phyloseq object

Examples

```
library(microbiome)
data(atlas1006)
validate(atlas1006)
```

Index