Package ‘theseus’

December 20, 2017

Type Package

Title Analysis and Visualization Tools for Microbial Community Data

Version 0.1.0

Description An approach to the visualization, analysis, and interpretation of (microbial) community composition data, especially those originating from amplicon sequencing. Analysis techniques include constrained and unconstrained ordination and visualizing taxonomic abundances and spatial patterns, among others. Methods intended to assist bioinformaticians and ecologists in selecting read trimming by quality scores and preprocessing/denoising of datasets are also provided.

License MIT + file LICENSE

URL `http://github.com/EESI/theseus`

BugReports `http://github.com/EESI/theseus/issues`

Encoding UTF-8

LazyData TRUE

Depends R (>= 3.3.0)

Imports dplyr, ggplot2 (>= 2.2.1), gridExtra, magrittr (>= 1.5), parallel (>= 3.4.1), phyloseq (>= 1.20.0), ShortRead (>= 1.34.0), splancs, tidyverse (>= 1.1.1), tidyr, vegan (>= 2.3.5), viridis (>= 0.4.0)

Suggests covr, knitr, rmarkdown, testthat

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Jacob Price [aut],
Stephen Woloszynek [cre, aut]

Maintainer Stephen Woloszynek <`sw424@drexel.edu`>

Repository CRAN

Date/Publication 2017-12-20 16:30:53 UTC

R topics documented:

`cohort_relabund` *Creates a relative abundance cohort plot*

Description

This function plots the relative abundance of taxa within a phyloseq object 'PS' according to thier pre-determined cohort memberships. See Details for more information.

Usage

```
cohort_relabund(PS, xvar = "SampleID", taxfill = "Phylum", comp1, comp2,
  comp1lab = c("Decreased Comp1", "No Change Comp1", "Increased Comp1"),
  comp2lab = c("Decreased Comp2", "No Change Comp2", "Increased Comp2"),
  justdata = FALSE, PSisRelAbund = FALSE)
```

Arguments

`PS`	(required) A phyloseq object.
`xvar`	(required) Variable in sample_data(PS) to be displayed on the x-axis. Defaults to 'SampleID'.
`taxfill`	Taxonomic level to display in plot. Defaults to 'Phylum'.
`comp1`	(required) First comparison (hence 'comp1') object of 'DESeqResults' class, or a dataframe with similar structure. In the case that a 'DESeqResults' class object is not being used, the object must contain a 'log2FoldChange' vector/column. Row names must be a subset (but not necessarily a proper subset) of taxa_names(PS). See Details for more information.
`comp2`	(required) Second comparison (hence 'comp2') object. Refer to documentation for 'comp1' for remaining details.
`comp1lab`	Labels for comparison 1. Defaults to c('Decreased Comp1', 'No Change Comp1','Increased Comp1').

`comp2lab` Labels for comparison 2. Defaults to c(’Decreased Comp2’, ’No Change Comp2’,’Increased Comp2’).

`justdata` Return only the data table (no plot). Defaults to FALSE.

`PSisRelAbund` Does the PS object contain compositional (relative abundance) taxa counts? Defaults to FALSE.

Details

General: The results from a single pairwise comparison, such as pre- and post-treatment, carried out with the DESeq2 package can be plotted or read and interpreted in tabular form with relative ease. When two pairwise comparisons are being performed, interpreting the results becomes more difficult. This function is intended to assist with interpreting the results from multiple differential abundance analyses carried out with the DESeq2 r-package. This function takes a phyloseq object (’PS’) and two ’DESeqResults’ objects (’comp1’, ’comp2’) and plots the relative abundance of taxa within ’PS’, partitioning the taxa according to their membership to one the 9 possible cohort combinations determined by their values specified within ’comp1’ and ’comp2’.

Approach: The `DESeq` function carries out differential abundance testing and produces a ’DESeqDataSet’ object. The `results` function can be used to access the results and create a ’DESeqResults’ object, which is a subclass of DataFrame. Note that the ’alpha’ parameter for `results` can be used to specify the significance level of the test being performed. NOTE: Testing with DESeq2 must be carried out, and the non-significant taxa should be removed from the ’comp1’ and ’comp2’ objects before using this function. Using the log2FoldChange columns in ’comp1’ and ’comp2’ this function identifies which taxa decrease, do not change, or increase over course of both comparisons. Because there are three options for both comparisons there are 3^2=9 possible combinations, or cohorts, which an OTU may fall into. These cohort assignments are used when plotting the relative abundance plot.

Value

A ggplot object.

See Also

`plot_bar` `transform_sample_counts` `DESeq` `results` `DESeqDataSet`

Examples

```
## Not run:
cohort_relabund(
  PS=prune_samples(sample_data(WWTP_Impact)$site %in% c(1,2,3,4),
                   WWTP_Impact),
  comp1=sigtab,
  comp2=sigtab.2vs3,
  comp1lab=c('Decreased at Effluent',
             'No change at Effluent',
             'Increased at effluent'),
  comp2lab=c('Decreased btwn plants',
             'No change btwn plants',
             'Increased btwn plants'))
```

```
## End(Not run)
```

constord *Plots constrained ordination results*

Description

Function ’conord’ (constrained ordination) carries out constrained ordination on a phyloseq object and plots the results as a ggplot2 object. Constrained correspondence analysis (’CCA’) and redundancy analysis (’RDA’) are the two methods currently implemented within this function.

Usage

```
constord(PS, formula, method = c("CCA", "RDA"), facets, scaling = 2,
  tax_level = "Phylum", tax_n = 7)
```

Arguments

PS	(required) A phyloseq object.
formula	(required) Right-hand side of the model formula starting with a tilde (’~’) and should not be placed in quotes. The current version requires at least two (2) constraining variables (see Details).
method	Constrained ordination method to be applied. User may choose Constrained Correspondence Analysis (CCA) or Redundancy Analysis (RDA). Defaults to CCA.
facets	Variable in sample_data(PS) to facet the plot by. Statement starts with a tilde (’~’) and should not be placed in quotes.
scaling	Scaling for species and site/sample scores in biplot. Options are the same as those found in the scores function: "species" scaling (1) or ’site’ scaling (2). The user should designate the appropriate scaling for thier intended analysis. Further information regarding scaling can be found in the Details below. Defaults to 2.
tax_level	Taxonomic level to represent species composition using color. Defaults to ’Phylum’.
tax_n	The number of taxonomic groups to identify using color (at the taxonomic level ’tax_level’). The most abundant tax_n will be selected. All other taxonomic groups will be collapsed into an additional ’Other’ category for visualization. Defaults to 7.

Details

General: The current implementation of this function displays the first two constrained ordination axes. As such, the 'formula' argument must contain two or more constraining variables to return a valid plot; an error will be returned otherwise. Legendre and Legendre (1998, p. 587-592, 597-600, Table 11.1-11.5) provide thorough discussion on the constrained and unconstrained axes that result from constrained ordination methods. Selection of axes to plot, constrained or unconstrained, is a planned feature for a future release.

Comparison with phyloseq::plot_ordination: There are several differences between 'constord' and `plot_ordination`, and they each have their own strengths. The highlights of 'constord' are:

- constraining variables are included in the 'constord' plot, a feature not currently present in phyloseq's approach, but still possible with some extra coding.
- 'constord' has argument 'scaling' which allows the user to select whether species scaling (1) or site scaling (2) is used when returning the scores to be plotting. Currently, *phyloseq::plot_ordination* returns site scaling (2). The choice of scaling is important and should be selected depending upon whether the goal is to compare the arrangement of sites or species (see Scalng section below).
- The aspect ratio of the ordination plots themselves are scaled according the ordination's eigenvalues to more accurately represent the distances between sites/samples, as described by Callahan et. al. (2016). Once again, this is easily included in `plot_ordination`

Scaling: Species scaling (1) results in a distance biplot. The distance biplot is intended to enable the user to interpret the relationships between sites/samples. Site scaling (2) results in a correlation biplot. The correlation biplot enables the user to interpret the correlation between descriptors (species) within the ordination. Positions of sites/samples are not approximations of thier true locations; use species scaling (1) to interpret site/samples. A complete discussion of the implications of scaling (and interpretation of the ordination results) is provided in Legendre and Legendre (1998, p. 403-404, 585-587).

Value

A ggplot object.

References

Legendre, P. and Legendre, L. (1998) Numerical Ecology. 2nd English ed. Elsevier. Callahan BJ, Sankaran K, Fukuyama JA et al. Bioconductor Workflow for Microbiome Data Analysis: from raw reads to community analyses [version 2; referees: 3 approved]. F1000Research 2016, 5:1492 (doi: 10.12688/f1000research.8986.2)

See Also

`ordinate` `plot_ordination` `cca` `rda` `scores`

Examples

```
## Not run:
library(theseus)
data('WWTP_Impact')
```

```
p.co <- constord(PS=WWTP_Impact,
                 formula=~ log_NO3N + log_PO4,
                 method='RDA', facets=Position~Location, scaling=2)
p.co

## End(Not run)
```

envtoverlay *Environmental variable fitting to unconstrained ordination diagrams*

Description

Fits environmental variables as vectors (via envfit) and smooth surfaces (via ordisurf) to an ordination diagram. The figure is faceted if multiple variables are specified.

Usage

```
envtoverlay(PS, covariates, ordmet = "PCA")
```

Arguments

PS	(required) A phyloseq object.
covariates	(required) A character vector of covariates present in the phyloseq objects sample_data().
ordmet	Ordination method. Options are Principal Component Analysis ("PCA") or Correspondence Analysis ("CA"). Defaults to "PCA".

Value

A ggplot object.

See Also

rda cca envfit ordisurf

Examples

```
## Not run:
library(theseus)
library(phyloseq)
library(ggplot2)
data('WWTP_Impact')
cv <- c('log_NO3N', 'log_PO4')
p.eo <- envtoverlay(WWTP_Impact, covariates=cv)
p.eo

## End(Not run)
```

prev *Create prevalence vs abundance plot*

Description

Function 'prev' (prevalence plot) tabulates the prevalence and abundance of each taxa in a phyloseq object and plots the results as a ggplot object. This may assist the user in determining what filtering and preprocessing steps should be taken regarding the removal of low count taxa.

Usage

```
prev(PS, taxon, n_taxa = 10, abund_threshold = 3,
  prev_threshold = ceiling(0.05 * phyloseq::nsamples(PS)))
```

Arguments

`PS`	(required) A phyloseq object.
`taxon`	Taxonomic level to be displayed. Defaults to 'Phylum'.
`n_taxa`	The number of taxa to display at the given 'taxon' level. The most abundant taxa at that group are selected. Defaults to 10.
`abund_threshold`	User selected value for the lowest acceptable total abundance. Defaults to 3.
`prev_threshold`	User selected value for the lowest acceptable prevalence. Defaults to 5 rounded up to the nearest integer.

Details

Low count taxa are often filtered from OTU tables to reduce possible error or noise. Examination of the raw (unfiltered) OTU table should be carried out to ensure that appropriate thresholds for prevalence (number of samples a taxa was observed in) and abundance (the total number of times a taxa was observed) are being selected. Function 'prev' plots each taxa according to thier prevalence and abundance within the dataset.

Value

A ggplot object.

References

Callahan BJ, Sankaran K, Fukuyama JA et al. Bioconductor Workflow for Microbiome Data Analysis: from raw reads to community analyses. F1000Research 2016, 5:1492 (doi: 10.12688/f1000research.8986.2)
Perraudeau F, Risso D, Street K et al. Bioconductor workflow for single-cell RNA sequencing: Normalization, dimensionality reduction, clustering, and lineage inference. F1000Research 2017, 6:1158 (doi:10.12688/f1000research.12122.1)

Examples

```
## Not run:
library(theseus)
data('WWTP_Impact')
p.prev <- prev(WWTP_Impact, taxon="Phylum", n_taxa=10)
p.prev

## End(Not run)
```

---

pstoveg_otu — *converts the otu_table slot of a phyloseq object to a vegan-compatible matrix*

Description

physeq2veg_otu is a helper function intended to convert the species/taxa count slot of a phyloseq object to a vegan-friendly matrix. This function ensures that sites/samples are rows and species are columns.

Usage

```
pstoveg_otu(PS)
```

Arguments

PS	(required) a phyloseq object

Value

A matrix containing a phyloseq object's otu_table slot.

See Also

phyloseq-class otu_table-class otu_table

Examples

```
## Not run:
library(theseus)
library(phyloseq)
data(WWTP_Impact, package='theseus')
dim(otu_table(WWTP_Impact))
taxa_are_rows(WWTP_Impact)
otu <- pstoveg_otu(WWTP_Impact)
dim(otu)

data(GlobalPatterns, package='phyloseq')
dim(otu_table(GlobalPatterns))
```

```
taxa_are_rows(GlobalPatterns)
otu.gp <-pstoveg_otu(GlobalPatterns)
dim(otu.gp)

# move transformed OTU table back to phyloseq
wwtp <- WWTP_Impact
otu.ra <- vegan::decostand(otu, method='total')
otu_table(wwtp) <- otu_table(otu.ra,
                             taxa_are_rows = taxa_are_rows(WWTP_Impact))

## End(Not run)
```

pstoveg_sample *Converts sample_data slot of a phyloseq object to vegan-compatible matrix*

Description

physeq2veg_otu is a helper function intended to convert the species/taxa count slot of a phyloseq object to a vegan-friendly matrix.

Usage

```
pstoveg_sample(PS)
```

Arguments

PS	(required) a phyloseq object

Value

A matrix containing a phyloseq object's otu_table slot.

See Also

phyloseq-class otu_table-class otu_table

Examples

```
## Not run:
library(phyloseq)
data("GlobalPatterns")
# inspect otu_table()
dim(otu_table(GlobalPatterns))
str(otu_table(GlobalPatterns))
taxa_are_rows(GlobalPatterns)
gp.otu <- physeq2veg_otu(GlobalPatterns)
dim(gp.otu)
str(gp.otu)
```

```
## End(Not run)
```

pstoveg_sd — *converts the sam_data slot of a phyloseq object to a vegan-compatible matrix*

Description

physeq2veg_sd is a helper function intended to convert the sample data slot of a phyloseq object to a vegan-friendly matrix.

Usage

```
pstoveg_sd(PS)
```

Arguments

PS	(required) a phyloseq object

Value

A matrix containing a phyloseq object's sam_data slot.

See Also

phyloseq-class sample_data-class sample_data

Examples

```
## Not run:
library(theseus)
library(phyloseq)
data(WWTP_Impact, package='theseus')
dim(sample_data(WWTP_Impact))
sampdat <- pstoveg_sd(WWTP_Impact)
dim(sampdat)

data(GlobalPatterns, package='phyloseq')
dim(sample_data(GlobalPatterns))
sampdat.gp <-pstoveg_sd(GlobalPatterns)
dim(sampdat.gp)

# move altered sample data back to phyloseq
sampdat.altered <- sampdat
sampdat.altered$TotDisP_PercentMax <- vegan::decostand(sampdat$TotDisP,
                                                       method='max')
sample_data(wwtp) <- as.data.frame(sampdat.altered)
```

```
## End(Not run)
```

qualcontour *Create read quality color contour plots*

Description

This function generates a 2-D color/contour map representing the average quality scores by location (read cycle number) for a designated percentile. It is intended to assist the user with deciding where trimming should be performed.

Usage

```
qualcontour(f_path, r_path, idx, percentile = 0.25, amp_length, min_overlap,
  n_samples = 12, q = c(25, 30, 35), bins = 50, nc = 1,
  seed = sample.int(.Machine$integer.max, 1), verbose = FALSE)
```

Arguments

f_path	(required) A character vector locating the forward read (Read 1) .fastq files
r_path	(required) A character vector locating the reverse read (Read 2) .fastq files
idx	Indexes (within f_path and r_path) identifying specific .fastq files to be used for analysis
percentile	The percentile to be targeted . Defaults to .25 (i.e. the first quartile).
amp_length	Intra-primer amplicon length. Calculated distance in base-pairs between primers. Used to determine region of no overlap. Both 'amp_length' and 'min_overlap' must be provided for these calculations.
min_overlap	The minimum amount of overlap between the two reads. Used to determine region of no overlap. Both 'amp_length' and 'min_overlap' must be provided for these calculations.
n_samples	Integer indicating the number of samples to include in the visualization. Defaults to 12.
q	A numeric vector designating Phred quality scores to be represented on the plot. Defaults to 25, 30, and 35.
bins	Integer designating the number of bins each read should be separated into. For example, visualizing a 250 bp read with 50 bins would imply that each bin represents 5 cycles/bp. Increasing the number of bins improves granularity at the cost of memory and processing speed. Defaults to 50.
nc	The number of cores to use when multithreading. Defaults to 1.
seed	An integer value to be used when randomly selecting the subset of samples to be visualized.
verbose	If set to TRUE, provides verbose output. Defaults to FALSE.

Details

qualcontour's (quality contour) two required arguments are character vectors of the file paths for forward ('f_path') and reverse ('r_path') reads. qualcontour tabulates the distribution of quality scores at each read cycle for the forward and reverse reads independently and then averages (arithmetic mean) the quality scores for each (forward/reverse) cycle combination. These values are then plotted as a ggplot2 object. Users can (re)run 'qualcontour' with different 'percentile' values to visualize how the quality scores varies in shape. `plotQualityProfile` in the 'dada2' package provides an elegant way of looking at the quality profiles for the forward or reverse reads.

Value

A ggplot object with the following attributes:

idx Samples used to generate the plot.

amp_length Value for amp_length used to generate the plot.

min_overlap Value for min_overlap used to generate the plot.

seed Seed used to select the samples used to generate the plot.

See Also

`qa` `plotQualityProfile`

Examples

```
## Not run:
library(theseus)
library(ggplot2)
fns <- sort(list.files(file.path(system.file(package='theseus'),
            '/testdata/'), full.names=TRUE))
f_path <- fns[grepl('R1.fastq.gz', fns)]
r_path <- fns[grepl('R2.fastq.gz', fns)]
p.qc <- qualcontour(f_path, r_path, n_samples=2, verbose=TRUE,
                    percentile=.25, nc=1)
p.qc
p.qc + geom_hline(yintercept=175) + geom_vline(xintercept=275)

## End(Not run)
```

---

sigtab *Differential abundance data*

Description

Differential abundance data

Usage

```
sigtab
```

Format

A 137x13 dataframe containing DESeq2 differential abundance data.

sigtab.2vs3 *Differential abundance data*

Description

Differential abundance data

Usage

```
sigtab.2vs3
```

Format

A 205x13 dataframe containing DESeq2 differential abundance data.

WWTP_Impact *Freshwater stream microbiome data*

Description

WWTP_Impact is a phyloseq object containing 16S rDNA amplicon sequencing data from samples collected at 6 sites on 2 days (12 total samples) along Wissahickon Creek and Sandy Run in southeastern Pennsylvania, USA. Sequencing data was processed as described in Price et. al. (2017), and combined with chemical data to create a phyloseq object. The raw data and scripts used to generate this phyloseq object (as well as the phyloseq object itself) can be obtained from the author's GitHub repository (see url below).

Usage

```
WWTP_Impact
```

Format

A phyloseq object containing sample_data, otu_table, tax_table, and phy_tree slots.

Source

https://github.com/JacobRPrice/WWTP_Impact_on_Stream

References

Price, J. R., Ledford, S. H., Ryan, M. O., Toran, L., Sales, C. M. (2017). Wastewater treatment plant effluent introduces recoverable shifts in microbial community composition in receiving streams. Sci. Total Environ. doi:10.1016/j.scitotenv.2017.09.162.

Index