Package: BRCore (via r-universe)

May 20, 2026

Title A Unified Framework for Identification and Ecological Interpretation of Microbial Data from Bioenergy Research Centers

Version 2.0.7.9000

Description A unified framework for identification and ecological interpretation of core microbiomes across time and space, enhancing robustness and reproducibility in microbiome data analysis. 'BRCore' implements the workflow proposed by Shade and Stopnisek (2019) and incorporates additional rarefaction steps. The proposed workflow aims to identify persistent microbiomes using abundance-occupancy distributions and neutral community model fitting. For more details on abundance-occupancy distributions see Shade A, Stopnisek N (2019) <doi:10.1016/j.mib.2019.09.008>, for neutral models, see Sloan et al. (2006) <doi:10.1111/j.1462-2920.2005.00956.x> and Burns et al. (2015) <doi:10.1038/ismej.2015.142>.

Depends R (>= 4.4)

Imports cli (>= 3.6.5), dplyr (>= 1.1.4), ggplot2 (>= 4.0.1), ggpubr (>= 0.6.2), ggrepel (>= 0.9.6), Hmisc (>= 5.2.4), magrittr (>= 2.0.4), minpack.lm (>= 1.2.4), phyloseq, rlang (>= 1.1.6), scales (>= 1.4.0), stats4, tibble (>= 3.3.0), tidyr (>= 1.3.1), vegan (>= 2.7.2)

Suggests knitr (>= 1.50), rmarkdown (>= 2.30), testthat (>= 3.2.3), tidyverse (>= 2.0.0), viridis (>= 0.6.5)

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

LazyData true

VignetteBuilder knitr

URL https://github.com/germs-lab/BRCore,
https://www.germslab.org/BRCore/

BugReports https://github.com/germs-lab/BRCore/issues

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev zlib1g-dev

Repository https://germs-lab.r-universe.dev

Date/Publication 2026-05-20 17:37:12 UTC

RemoteUrl https://github.com/germs-lab/brcore

RemoteRef HEAD

RemoteSha b08c3c183ce71153685ef9ec9bba254a8dca6cb1

Contents

add_rarefaction_metrics . . . 2
bcse . . . 4
bean . . . 4
fit_neutral_model . . . 5
identify_core . . . 6
mimulus . . . 9
multi_rarefy . . . 10
plot_abundance_occupancy . . . 11
plot_core_distribution . . . 12
plot_identified_core . . . 14
plot_neutral_model . . . 15
plot_rarefaction_metrics . . . 16
plot_variance_propagation . . . 17
sncm.fit . . . 18
switchgrass . . . 20
switchgrass_core . . . 21
update_otu_table . . . 21

Index **23**

`add_rarefaction_metrics`
Calculate and append pre-rarefaction statistics to microbiome data

---

Description

This function adds read count, singleton count, Good's coverage, and marks outlier samples to a `phyloseq` object or `data.frame` based on the OTU/ASV abundance table.

Usage

```
add_rarefaction_metrics(data)
```

Arguments

data	A phyloseq object or a data.frame with samples as rows and taxa as columns.

Details

About Good's coverage. Initially developed by Alan Turing and I.J. Good during their cryptographic analyses in World War II, it was later adopted by ecologists, particularly in microbial diversity studies, to assess the completeness of a sample's representation of the overall community. It's calculated as 1 - (F1/N), where F1 is the number of OTUs (Operational Taxonomic Units) represented by only one individual (singletons) and N is the total number of individuals in the sample. For example, a Good's coverage of 0.95, means that 5% of the reads in that sample are from OTUs that appear only once.

Value

The same object (phyloseq or data.frame) with new columns:

- read_num
- singleton_num
- goods_cov
- outlier

See Also

plot_rarefaction_metrics() for visualizing these metrics, and multi_rarefy() for performing rarefaction on a phyloseq object.

Examples

```
library(phyloseq)
library(BRCore)

# Adding metrics to a "phyloseq" object
bcse_metrics <- add_rarefaction_metrics(data = bcse)
sample_data(bcse_metrics)|>
head(10)


# Adding metrics to a "data.frame" count table object
bcse_otutable <- as.data.frame(
  as.matrix(otu_table(bcse))
)

bcse_otutable_metrics <- add_rarefaction_metrics(
  data = bcse_otutable
)
bcse_otutable_metrics[
  head(seq_len(nrow(bcse_otutable_metrics)), 10),
  tail(seq_len(ncol(bcse_otutable_metrics)), 20)
]
```

---

bcse *16S amplicon dataset from the GLBRC Biofuel Cropping System Experiment (BCSE) at Michigan State University, Kellogg Biological Station*

Description

A phyloseq containing 2861 97% sequence similarity OTUs (Operational Taxonomic Units) across 50 samples. Samples are for the leaf microbiome only.

Usage

```
data("bcse")
```

Format

An object class "phyloseq".

Source

Internal test dataset.

References

Haan, N. L., Benucci, G. N. M., Fiser, C. M., Bonito, G., & Landis, D. A. (2023). Contrasting effects of bioenergy crops on biodiversity. Science Advances, 9(38), eadh7960. doi:10.1126/sciadv.adh7960

---

bean *16S amplicon dataset from common bean (Phaseolus vulgaris)*

Description

A phyloseq containing 24885 ASVs (Amplicon Sequence Variants) across 30 samples.

Usage

```
data("bean")
```

Format

An object class "phyloseq".

Source

Internal test dataset.

References

Shade A, Stopnisek N (2019) Abundance-occupancy distributions to prioritize plant core microbiome membership. Current Opinion in Microbiology, 49:50-58 doi:10.1016/j.mib.2019.09.008

`fit_neutral_model` *Fit a Neutral Model to Microbial Community Data*

Description

The function fits the neutral distribution model developed by Sloan et al. (2006), and implemented in R by Burns et al. (2015), to an OTU/ASV table and returns several goodness of fit statistics alongside a data.frame with predicted occurrence frequencies for each OTU/ASV based on their abundance in the metacommunity for plotting the abundance-occupancy distribution. In addition, this function identified core and not-core OTU/ASVs that are neutral (i.e. stochastically or randomly distributed) above (i.e. interpreted as deterministically or predictably selected) and below (i.e. interpreted as dispersally limited) the model predictions.

Usage

```
fit_neutral_model(otu_table, core_set, abundance_occupancy)
```

Arguments

`otu_table`	A community count matrix (or OTU table) with samples as rows and OTU/ASVs as columns..
`core_set`	Character vector of core OTU/ASVs IDs (must match column names of `otu_table`).
`abundance_occupancy`	data.frame (or tibble) with OTU/ASVs names, occupancy (`otu_occ`), and mean relative abundance (`otu_rel`) as generated by identify_core().

Value

A list with:

- `goodness_of_fit`: one-row tibble of summary stats (plus `above.pred`, `below.pred`) - `model_prediction`: tibble with `abundance_occupancy` joined to per-taxon predictions

References

Sloan, W. T., Lunn, M., Woodcock, S., Head, I. M., Nee, S., & Curtis, T. P. (2006). Quantifying the roles of immigration and chance in shaping prokaryote community structure. Environmental Microbiology, 8(4), 732–740. doi:10.1111/j.1462-2920.2005.00956.x

Burns, A. R., Stephens, W. Z., Stagaman, K., Wong, S., Rawls, J. F., Guillemin, K., & Bohannan, B. J. M. (2016). Contribution of neutral processes to the assembly of gut microbial communities in the zebrafish over host development. The ISME Journal, 10(3), 655–664. doi:10.1038/ismej.2015.142

Shade A, Stopnisek N (2019) Abundance-occupancy distributions to prioritize plant core microbiome membership. Current Opinion in Microbiology, 49:50-58 doi:10.1016/j.mib.2019.09.008

See Also

plot_neutral_model()

Examples

```
library(BRCore)
data("switchgrass_core", package = "BRCore")

switchgrass_core_fit <- fit_neutral_model(
  otu_table = switchgrass_core$otu_table,
  core_set = switchgrass_core$increase_core,
  abundance_occupancy = switchgrass_core$abundance_occupancy
)

str(switchgrass_core_fit)
switchgrass_core_fit$goodness_of_fit
switchgrass_core_fit$model_prediction |>
head()
```

`identify_core` *Identify Core Microbiome Using Bray-Curtis Similarity biological samples. Core taxa are selected using either a "last % increase" or "elbow" method implementing the method developed by Shade and Stopnisek (2019) Curr Opin Microbiol, see below for details.*

Description

Identify Core Microbiome Using Bray-Curtis Similarity biological samples. Core taxa are selected using either a "last % increase" or "elbow" method implementing the method developed by Shade and Stopnisek (2019) Curr Opin Microbiol, see below for details.

Usage

```
identify_core(
  physeq_obj,
  rarefied_list = NULL,
  priority_var,
  increase_value = 0.02,
  abundance_weight = 0,
  max_otus = NULL,
  depth_level = 1000,
  seed = NULL
)
```

Arguments

physeq_obj	A phyloseq object with at least otu_table and sample_data. If the phyloseq object is already rarefied (i.e., all samples have the same sequencing depth), you can omit rarefied_list and the function will automatically use the OTU table from physeq_obj as a single iteration. If the data is not rarefied, you must provide a rarefied_list generated by multi_rarefy. Alternatively you can use update_otu_table to replace the OTU table with a rarefied version before running identify_core.
rarefied_list	A list of data frames, each representing a rarefied OTU table (taxa x samples) generated by multi_rarefy. Required if physeq_obj is not already rarefied. If physeq_obj is rarefied, this can be omitted and the function will use the OTU table from physeq_obj as a single iteration.
priority_var	The column name in the sample_data (e.g. sampling_date", "site") that is used for prioritizing the core microbiome.
increase_value	Increase value (numeric, scalar) used in the calculation (default 0.02) for "increase". The "elbow" is always calculated and returned as elbow_core (see below for details).
abundance_weight	Numeric in [0,1]; how much to weight mean relative abundance in the ranking score. 0 (default) uses occupancy/composite only. 1 ranks purely by abundance. Values in between blend the two (e.g., abundance_weight = 0.3 gives 70% occupancy/composite + 30% abundance).
max_otus	Optional integer to limit analysis to the top N ranked OTUs. If NULL (default), all OTUs are analyzed. Useful for large datasets (>5000 OTUs)
depth_level	Integer. The sequencing depth used for normalization in Bray-Curtis calculations. If data is rarefied, this is automatically set to the rarefaction depth. For unrarefied data, samples with depth below this threshold are excluded from pairwise comparisons.
seed	Optional integer to set the RNG seed for reproducibility.

Details

The core set is defined using two separate methods:

The function rank OTU/ASVs by occupancy (optionally with abundance weighting: `rank_score = (1 - weight) * rank + weight * scaled_abundance`, where scaled_abundance is mean relative abundance rescaled to `[0,1]`). For each `k = 1..K`, recompute `S_k` as the mean Bray-Curtis similarity across all sample pairs using only the first `k` ranked OTUs; when `k = K`, this yields `S_K`, the value computed with all OTUs. Normalizing by `S_K` gives `C_k = S_k / S_K`.

The **elbow** is the point of diminishing returns: for each `k`, compare the average *left* slope `(S_k - S_1) / (k - 1)` to the average *right* slope `(S_K - S_k) / (K - k)`, and choose the `k` that maximizes `(left - right)`.

The **last percent Bray-Curtis increase** method uses the same accumulation curve, examine the multiplicative step when adding the `k-th` OTU: `Increase_k = S_k / S_{k-1}` (equivalently, `Increase_k = C_k / C_{k-1}`). Choose the largest `k` such that `Increase_k >= 1 + p`, where `p` is your chosen percent threshold (increase_value; recommended `p >= 0.02` or `2%`). This selects the final rank for which adding one more taxon still increases the explained Bray-Curtis similarity by at least `p`.

Value

A list with:

- `bray_curtis_ranked` tibble with `rank`, mean Bray-Curtis similarity across sample pairs (`MeanBC`) at each cumulative `rank`, normalized proportion (`proportionBC`), the multiplicative `IncreaseBC`, and the elbow metric (`elbow_slope_diffs`). (`proportionBC`), the multiplicative `IncreaseBC`, and the elbow metric (`elbow_slope_diffs`).
- `otu_ranked` tibble with ranked OTU/ASVs .
- `abundance_occupancy` tibble with OTU/ASVs names, occupancy (`otu_occ`), and mean relative abundance (`otu_rel`).
- `priority_var` character, the variable used for prioritizing the core.
- `elbow` core set identified by elbow method (integer).
- `bc_increase` core set identified by last % BC-increase (integer).
- `increase_value` increase value (numeric, scalar) used in the calculation (e.g. 0.02).
- `elbow_core` core OTU/ASVs using elbow method (character vector).
- `increase_core` core OTU/ASVs using last % BC-increase method (character vector).
- `otu_table` otu_table counts (otu x samples) used (data.frame).
- `sample_metadata` samples metadata (data.frame).
- `taxonomy_table` taxonomy if present (data.frame); otherwise NULL.

Dependencies

Requires **phyloseq**, **dplyr**, **tidyr**, **tibble**, **rlang**, and **vegan**.

References

Shade A, Stopnisek N (2019) Abundance-occupancy distributions to prioritize plant core microbiome membership. Current Opinion in Microbiology, 49:50-58 doi:10.1016/j.mib.2019.09.008

See Also

multi_rarefy(), plot_identified_core(), and plot_core_distribution()

Examples

```
library(BRCore)

# With rarefied data
res <- identify_core(
  physeq_obj     = switchgrass,
  priority_var   = "sampling_date",
  increase_value = 0.02,
  seed           = 091825
)

str(res)

# With unrarefied data (requires multi_rarefy step)
rarefied_list <- multi_rarefy(
  physeq_obj = bcse,
  depth_level = 1000,
  num_iter = 3,
  .as = "list",
  set_seed = 7642
)


res_rare <- identify_core(
  physeq_obj = bcse,
  rarefied_list = rarefied_list,
  priority_var = "Crop",
  increase_value = 0.02,
  seed = 091825
)

str(res_rare)
```

---

mimulus *16S amplicon dataset from yellow monkeyflower (Mimulus guttatus)*

Description

A phyloseq containing 11085 ASVs (Amplicon Sequence Variants) across 39 samples.

Usage

```
data("mimulus")
```

Format

An object class "phyloseq".

Source

Internal test dataset.

References

Shade A, Stopnisek N (2019) Abundance-occupancy distributions to prioritize plant core microbiome membership. Current Opinion in Microbiology, 49:50-58 doi:10.1016/j.mib.2019.09.008

`multi_rarefy` *Run rarefaction for microbiome count tables*

Description

This function performs rarefaction on a `phyloseq` object by randomly sub-sampling OTUs/ASVs within samples without replacement for a number of iterations specified by the user. Samples with fewer OTUs/ASVs than the specified `depth_level` are discarded.

Usage

```
multi_rarefy(
  physeq_obj,
  depth_level,
  num_iter = 100,
  .as = "list",
  set_seed = NULL
)
```

Arguments

`physeq_obj`	A `phyloseq` object containing an OTU/ASV table.
`depth_level`	An integer specifying the sequencing depth (number of OTUs/ASVs) to which samples should be rarefied.
`num_iter`	An integer specifying the number of iterations to perform for rarefaction.
`.as`	A character string indicating whether to return the results as a 3D array or as a list of data frames. If `"array"`, returns a 3D array with dimensions (samples x taxa x iterations). If `"list"`, returns a list of data frames, one for each iteration, with samples as rows and taxa as columns. (default = "list")
`set_seed`	An optional integer to set the random seed for reproducibility (default = NULL).

Value

A data frame with taxa as rows and samples as columns. The values represent the average sequence counts calculated across all iterations. Samples with less than `depth_level` sequences are discarded.

See Also

`update_otu_table()` for updating the OTU table in a `phyloseq` object and `vegan::rrarefy()` for the underlying rarefaction method used in this function.

Examples

```
library(BRCore)


# Example rarefaction (single iteration, single core to keep examples fast)
otu_table_rare <- multi_rarefy(
  physeq_obj = bcse,
  depth_level = 1000,
  num_iter = 10,
  .as = "list",
  set_seed = 7642
)

rowSums(otu_table_rare[[1]])
```

---

plot_abundance_occupancy

Plot Abundance-Occupancy Curve and Display the Core Taxa

Description

Creates a scatter plot showing the relationship between mean relative abundance and occupancy (occurrence frequency) of taxa, with core taxa highlighted.

Usage

```
plot_abundance_occupancy(core_result, core_set = "elbow")
```

Arguments

`core_result` A list object returned by `identify_core`, containing at minimum:

- `occupancy_abundance`: A data frame with columns `otu`, `otu_rel` (mean relative abundance), and `otu_occ` (occupancy).
- `elbow_core`: Character vector of OTU IDs identified as core using the "elbow" method.

- `increase_core`: Character vector of OTU IDs identified as core using the "increase" method.

`core_set` Character string specifying which core set to highlight. Must be either "elbow" or "increase" (Default elbow).

Details

The function creates a scatter plot where each point represents a taxon (i.e. ASV or OTU). Core taxa (as defined by the selected method) are shown in red, while non-core taxa are shown in grey. The plot uses a log10 scale for abundance to better visualize the full range of abundances typically found in microbiome data.

Value

A ggplot object showing the abundance-occupancy plot with core taxa highlighted in red and non-core taxa in grey. The x-axis shows log10-transformed mean abundance and the y-axis shows occupancy (0-1).

See Also

`plot_core_distribution()` and `identify_core()`

Examples

```
library(BRCore)

data("switchgrass_core", package = "BRCore")

p <- plot_abundance_occupancy(
  core_result = switchgrass_core,
  core_set = "increase"
)
print(p)
```

---

plot_core_distribution

Plot Core Taxa Occupancy Across Metadata Groups

Description

Creates a plot showing core taxa (i.e. OTUs/ASVs) occupancy patterns across a grouping variable.

Usage

```
plot_core_distribution(
  core_result,
  core_set = "elbow",
  group_var = "Crop",
  plot_type = c("bar", "line", "heatmap")
)
```

Arguments

core_result	A list object returned by identify_core, containing at minimum: • otu_table: A data frame with ASV/OTUs as rows and samples as columns. • metadata: A data frame with samples as rows and grouping variables as columns. • otu_ranked: A data frame with ranked taxa containing: – otu: A column with taxa names. – rank: A column with the rank for each taxon. • elbow_core: Character vector of OTU IDs identified as core using the elbow method. • increase_core: Character vector of OTU IDs identified as core using the increase method.
core_set	Which core set to plot: "elbow" (default) or "increase".
group_var	Metadata column for bar coloring. Default: "sampling_date".
plot_type	Allows selection of 3 different plot types: bar, line, or heatmap.

Value

A ggplot2 object that can be further customized.

See Also

identify_core(), plot_abundance_occupancy(), and plot_identified_core()

Examples

```
library(BRCore)
data("switchgrass_core", package = "BRCore")

p <- plot_core_distribution(
  core_result = switchgrass_core,
  core_set = "increase",
  group_var = "sampling_date",
  plot_type = "bar"
)
print(p)
```

plot_identified_core *Plot Bray-Curtis increase over ranked OTU/ASVs*

Description

Visualize the cumulative normalized mean Bray-Curtis increase returned by identify_core(), over ranked OTU/ASVs and shows cutoff points for elbow percent increase methods.

Usage

```
plot_identified_core(
  bray_curtis_ranked,
  elbow,
  lastCall,
  increase_value = 0.02,
  dataset_name = NULL
)
```

Arguments

bray_curtis_ranked	A tibble as returned by identify_core()$bray_curtis_ranked.
elbow	The number of OTU/ASVs identified by the elbow method (Integer).
lastCall	The number of OTU/ASVs identified by the last percent Bray-Curtis increase method (Integer).
increase_value	The percent increase value in decimal (e.g. 0.02) used for the Bray-Curtis increase method.
dataset_name	Optional character string. When provided, it is prepended to the plot title (e.g. "Switchgrass"). Default NULL (no prefix).

Details

The function converts rank to integers and zooms the x-axis to the first 1.2 * lastCall ranks. Label positions are computed dynamically from the observed proportionBC range to avoid overlap.

Value

A list containing: 1) df_for_plot, a data frame used for plotting, and 2) plot_identified_core, a ggplot object visualizing the Bray-Curtis increase with annotated cutoff points.

See Also

identify_core()

identify_core

Examples

```
library(BRCore)
data("switchgrass_core", package = "BRCore")

p <- plot_identified_core(
  bray_curtis_ranked = switchgrass_core$bray_curtis_ranked,
  elbow = switchgrass_core$elbow,
  lastCall = switchgrass_core$bc_increase,
  increase_value = switchgrass_core$increase_value
)
print(p)
```

`plot_neutral_model` *Plot a fitted Neutral Model to Microbial Community Data*

Description

This function plots ASV/OTUs log abundances into a fitted neutral model of microbial abundance-occupancy distribution. ASV/OTUs that are *Core*, *As predicted* (i.e. Neutral), *Below* and *Above* model predictions are drawn with distinct point colors, see details.

Usage

```
plot_neutral_model(fit_result)
```

Arguments

`fit_result`	A list-like object returned by fit_neutral_model().

Details

Points are split into four groups for display:

- **Not core (as predicted)** – background points;
- **Core (as predicted)** – core taxa whose occupancy matches the model;
- **Core (above prediction)** – core taxa above the 95\
- **Core (below prediction)** – core taxa below the 95\

Value

A ggplot2 object.

A ggplot object with:

- *x*-axis: `log10(mean abundance)` (`log10(otu_rel)`);
- *y*-axis: `Occupancy` (`otu_occ`);
- Points colored by membership/fit class as described above;

- Neutral-model curve (solid) with 95\
- An inset white label reporting R^2 and m taken directly from `fit_result$goodness_of_fit`.

The function does not recompute statistics; it only visualizes the supplied predictions and metrics.

See Also

`fit_neutral_model`

Examples

```
library(BRCore)
data("switchgrass_core", package = "BRCore")

switchgrass_core_fit <- fit_neutral_model(
  otu_table = switchgrass_core$otu_table,
  core_set = switchgrass_core$increase_core,
  abundance_occupancy = switchgrass_core$abundance_occupancy
)

p <- plot_neutral_model(switchgrass_core_fit)
print(p)
```

---

plot_rarefaction_metrics

Plot pre-rarefaction diagnostics

Description

This function creates a 6-panel diagnostic plot showing sequencing depth, Good's coverage, and outlier behavior based on a data frame or a `phyloseq` object with rare stats already added via `add_rare_stats()`.

Usage

```
plot_rarefaction_metrics(data)
```

Arguments

`data`	Either a `phyloseq` object or a `data.frame` that includes columns `read_num`, `goods_cov`, and `outlier`.

Value

A `ggarrange` object with six plots.

See Also

`add_rarefaction_metrics()`

Examples

```
library(phyloseq)
library(BRCore)

# Add rarefaction metrics to the phyloseq object
bcse_metrics <- add_rarefaction_metrics(bcse)

# Plot the rarefaction diagnostics
plot_rarefaction_metrics(bcse_metrics)

# You can also pass a data frame directly if you have
# pre-computed read_num, goods_cov, and outlier columns
sample_data_df <- data.frame(sample_data(bcse_metrics))
plot_rarefaction_metrics(sample_data_df)
```

`plot_variance_propagation` *Variance propagation diagnostic for rarefaction*

Description

This function evaluate the variance between rarefaction iterations from `multi_rarefy()` by visually comparing raw vs. rarefied alpha diversity metrics calculated at each iterations. It is possible to plot observed richness (q=0), Shannon diversity (q=1), or Simpson diversity (q=2) by setting the `q` parameter to "richness" or `q = 0`, "shannon" or `q = 1`, or "shannon" or `q = 2`. The plot is faceted by method (raw vs rarefied) and colored by a specified grouping variable from the sample data.

Usage

```
plot_variance_propagation(
  physeq_obj,
  rarefied,
  q = 0,
  group_var,
  group_color,
  convert_to_factor = FALSE
)
```

Arguments

`physeq_obj`	Raw phyloseq object
`rarefied`	Output from multi_rarefy(). Either a list of dataframes or and array.
`q`	Hill number order (q = 0 for richness, q = 1 for Shannon, q = 2 for Simpson)
`group_var`	A grouping variable to use gor grouping as in the sample_data()
`group_color`	A color variable to use present in the sample_data()

convert_to_factor
: Logical. If TRUE, both group_var and group_color are coerced to factor before plotting, which is useful when those columns are numeric/continuous (e.g. dates, counts) but should be treated as discrete groups. When TRUE a discrete color scale (scale_color_viridis_d) is used; otherwise the continuous scale (scale_color_viridis_c) is used. Default FALSE.

Value

ggplot object comparing raw vs rarefied diversity distributions across iterations.

Examples

```
library(phyloseq)
library(BRCore)
# Example comparing hill q=1 between Poplar and Switchgrass plots
bcse_filt <- bcse |>
subset_samples(Crop %in% c("Poplar", "Switchgrass"))

bcse_rarefied_otutable_filt <-
 multi_rarefy(
       physeq_obj = bcse_filt,
       depth_level = 1000,
       num_iter = 10,
       .as = "list",
       set_seed = 7643
   )

plot_variance_propagation(
   physeq_obj   = bcse_filt,
   rarefied = bcse_rarefied_otutable_filt,
   q        = 1,
   group_var = "Crop",
   group_color = "Plot"
)
```

---

sncm.fit *Fit Sloan Neutral Community Model (SNCM)*

Description

The function implements the Sloan Neutral Community Model which predicts the occurrence frequency of taxa based on their relative abundance in the source pool and a migration parameter (m). It compares the neutral model against binomial and Poisson null models using various statistical measures.

Usage

```
sncm.fit(spp, pool = NULL, stats = TRUE, taxon = NULL)
```

Arguments

spp	A matrix or data frame where rows represent communities (samples) and columns represent species/taxa. Values should be abundances or counts.
pool	Optional. A matrix or data frame representing the source pool for calculating relative abundances. If NULL, the source pool is calculated from the spp matrix. Default is NULL.
stats	Logical. If TRUE, returns fit statistics including AIC, BIC, R-squared, and RMSE for model comparison. If FALSE, returns predicted vs. observed frequencies with confidence intervals. Default is TRUE.
taxon	Optional. A data frame containing taxonomic information to merge with results when stats = FALSE. Should have row names matching species names. Default is NULL.

Details

The model assumes neutral processes govern community assembly. Three models are compared: SNCM (beta distribution), binomial null model, and Poisson null model.

Value

If stats = TRUE, returns a data frame with model fit statistics including:

- m: Migration rate parameter from NLS fit
- m.ci: Confidence interval for m parameter
- m.mle: Migration rate from maximum likelihood estimation
- maxLL, binoLL, poisLL: Log-likelihood values for SNCM, binomial, and Poisson models
- Rsqr, Rsqr.bino, Rsqr.pois: R-squared values for each model
- RMSE, RMSE.bino, RMSE.pois: Root mean squared error for each model
- AIC, BIC: Information criteria for model selection
- N: Average number of individuals per community
- Samples: Number of samples/communities
- Richness: Number of taxa analyzed
- Detect: Detection limit (1/N)

If stats = FALSE, returns a data frame with observed and predicted frequencies along with confidence intervals for visualization.

References

Sloan WT, Lunn M, Woodcock S, Head IM, Nee S, Curtis TP. (2006) Quantifying the roles of immigration and chance in shaping prokaryote community structure. Environ Microbiol. 8(4):732-40. doi:10.1111/j.1462-2920.2005.00956.x

See Also

plot_neutral_model(), fit_neutral_model()

Examples

```
# Generate community data
set.seed(42)
n_samples <- 100
n_species <- 80

# Log-normal abundances
mean_abund <- rlnorm(n_species, meanlog = 2, sdlog = 1.5)
# Simulate community matrix with multinomial sampling
spp_data <- t(sapply(seq_len(n_samples), function(i) {
  rmultinom(
    1,
    size = sample(500:2000, 1),
    prob = mean_abund / sum(mean_abund)
  )[, 1]
}))
colnames(spp_data) <- paste0("Species_", seq_len(n_species))

fit_stats <- sncm.fit(spp_data, stats = TRUE)
predictions <- sncm.fit(spp_data, stats = FALSE)
```

---

switchgrass *16S amplicon dataset from Switchgrass (Panicum virgatum)*

Description

A phyloseq containing 706 97% sequence similarity OTUs (Operational Taxonomic Units) across 43 samples.

Usage

```
data("switchgrass")
```

Format

An object class "phyloseq".

Source

Internal test dataset.

References

Shade A, Stopnisek N (2019) Abundance-occupancy distributions to prioritize plant core microbiome membership. Current Opinion in Microbiology, 49:50-58 doi:10.1016/j.mib.2019.09.008

switchgrass_core *Identified core microbiome members for the switchgrass dataset*

Description

A phyloseq containing the identified core microbiome members for the switchgrass dataset. This object is used for BRCore function examples. It was generated by running the following on BRCore 2.0.2 (2026-04-30):

```
switchgrass_core <- identify_core(
  physeq_obj     = switchgrass,
  priority_var   = "sampling_date",
  increase_value = 0.02,
  seed           = 092825
)
```

Usage

```
data("switchgrass_core")
```

Format

An object class "phyloseq".

Source

Internal test dataset.

References

Shade A, Stopnisek N (2019) Abundance-occupancy distributions to prioritize plant core microbiome membership. Current Opinion in Microbiology, 49:50-58 doi:10.1016/j.mib.2019.09.008

update_otu_table *Add a rarefied otu_table to a phyloseq object*

Description

This function updates a phyloseq object by replacing its OTU/ASV table with a rarefied version produced by `multi_rarefy()`. The rarefied table can be a data frame, a list of data frames (`.as = "list"`), or a 3D array (`.as = "array"`). When providing a list or array, specify which iteration to use via the `iteration` parameter.

Usage

```
update_otu_table(physeq_obj, rarefied_otus, iteration = NULL)
```

Arguments

physeq_obj	A phyloseq object in which you want to add the rarefied OTU/ASV table.
rarefied_otus	A data frame, list of data frames, or 3D array output from multi_rarefy() containing the rarefied OTU/ASV tables.
iteration	Integer specifying which iteration to extract from a list or array. Required when rarefied_otus is a list or array. Ignored when rarefied_otus is a data frame.

Value

A phyloseq object.

See Also

multi_rarefy()

Examples

```
library(phyloseq)
library(BRCore)
data(GlobalPatterns, package = "phyloseq")

# List output
otu_list <-
  multi_rarefy(
    physeq_obj = GlobalPatterns,
    depth_level = 200,
    num_iter = 3,
    .as = "list",
    set_seed = 123
  )

# Extract iteration 2
rarefied_gp <- update_otu_table(GlobalPatterns, otu_list, iteration = 2)

# Array output
otu_array <-
  multi_rarefy(
    physeq_obj = GlobalPatterns,
    depth_level = 200,
    num_iter = 3,
    .as = "array",
    set_seed = 123
  )

# Extract iteration 1
rarefied_gp2 <- update_otu_table(GlobalPatterns, otu_array, iteration = 1)
```

Index

∗ **datasets**
  bcse, 4
  bean, 4
  mimulus, 9
  switchgrass, 20
  switchgrass_core, 21

add_rarefaction_metrics, 2
add_rarefaction_metrics(), *16*

bcse, 4
bean, 4

fit_neutral_model, 5, *16*
fit_neutral_model(), *15*, *19*

ggplot, *15*
ggplot2, *15*

identify_core, 6, *11*, *13, 14*
identify_core(), *5*, *12–14*

mimulus, 9
multi_rarefy, 10
multi_rarefy(), *3*, *9*, *22*

plot_abundance_occupancy, 11
plot_abundance_occupancy(), *13*
plot_core_distribution, 12
plot_core_distribution(), *9*, *12*
plot_identified_core, 14
plot_identified_core(), *9*, *13*
plot_neutral_model, 15
plot_neutral_model(), *6*, *19*
plot_rarefaction_metrics, 16
plot_rarefaction_metrics(), *3*
plot_variance_propagation, 17

sncm.fit, 18
switchgrass, 20
switchgrass_core, 21

update_otu_table, 21
update_otu_table(), *11*

vegan::rrarefy(), *11*