

# Package ‘HybridMicrobiomes’

January 20, 2025

**Type** Package

**Title** Analysis of Host-Associated Microbiomes from Hybrid Organisms

**Version** 0.1.1

**Description** A set of tools to analyze and visualize the relationships between host-associated microbiomes of hybrid organisms and those of their progenitor species. Though not necessary, installing the microViz package is recommended as a check for phyloseq objects. To install microViz from R Universe use the following command: `install.packages("microViz", repos = c(davidbarnett = "https://david-barnett.r-universe.dev", getOption("repos")))`. To install microViz from GitHub use the following commands: `install.packages("devtools")` followed by `devtools::install_github("david-barnett/microViz")`.

**License** GPL-2

**Encoding** UTF-8

**biocViews** Software, phyloseq

**Imports** rgl, compositions, rlang, PERMANOVA, graphics, stats, vegan, ks, ape, geometry, StereoMorph, phyloseq

**Suggests** microViz

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Benjamin Camper [aut],  
Zachary Lauglin [ctb],  
Daniel Malagon [ctb],  
Robert Denton [ctb],  
Sharon Bewick [aut, cre],  
National Science Foundation Division of Integrative Organismal Systems  
(award #2104605) [fnd],  
Clemson University Support for Early Exploration and Development (CU  
SEED) Grant [fnd]

**Maintainer** Sharon Bewick <sbewick@clemson.edu>

**Repository** CRAN

**Date/Publication** 2023-12-05 23:20:10 UTC

## Contents

FourHbootstrap	2
FourHbootstrapA	6
FourHcentroid	11
FourHcompare	12
FourHnull	14
FourHnullplane	16
FourHnullplaneD	18
FourHpreanalysis	19
FourHquaternary	22
FourHquaternaryC	23

<b>Index</b>	<b>26</b>
--------------	-----------

---

FourHbootstrap	<i>Bootstraps of the incidence-based 4H Index</i>
----------------	---

---

### Description

Calculates incidence-based 4H Indices for bootstrapped samples drawn from an object of class phyloseq containing host-associated microbial community data from both hybrid hosts and hosts of each parent species.

### Usage

```
FourHbootstrap(x, class_grouping, core_fraction, boot_no, sample_no,
  core_average_abundance = 0, core_max_abundance = 0,
  core_all_average_abundance = 0, core_all_max_abundance = 0,
  replace_hosts=FALSE, reads=NULL, rarefy_each_step=TRUE, seed=NULL, dist='Jaccard')
```

### Arguments

x	(Required) Host-associated microbial community data . This must be in the form of a phyloseq object and must contain an OTU abundance table with information on the host-associated microbial communities from both hybrid hosts and hosts of each progenitor taxon. The OTU abundance table should contain read counts, not fractional abundances.
class_grouping	(Required) A numerical vector identifying the host class of each host from the OTU table. This vector must be ordered according to the OTU abundance table, and must use the following notation: Progenitor One = 1, Hybrids = 2, Progenitor Two = 3.
core_fraction	(Required) The fraction of hosts that a microbial taxon must be found on to be considered part of a host class's 'core' microbiome
boot_no	(Required) The desired number of bootstrap samples
sample_no	(Required) The number of hosts of each host class that should be selected to generate each bootstrap sample. This number should be less than the minimum number of hosts in any given class.

core_average_abundance	The minimum average abundance at which a microbial taxon must be found across all hosts within a class to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum average abundance)
core_max_abundance	The minimum abundance at which a microbial taxon must be found on at least one host within a class to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum abundance on any single host)
core_all_average_abundance	The minimum average abundance at which a microbial taxon must be found across all hosts (regardless of class) to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum average abundance)
core_all_max_abundance	The minimum abundance at which a microbial taxon must be found on at least one host (regardless of class) to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum abundance on any single host)
replace_hosts	Whether or not to replace hosts during bootstrapping. The default is FALSE (i.e., all hosts in a given bootstrap are unique).
reads	The number of reads to rarefy to for each microbiome sample. The default is to use the number of reads in the sample with the lowest number of reads. Any user-specified value must be lower than the number of reads in the sample with the lowest number of reads.
rarefy_each_step	Whether or not to rarefy each bootstrap sample separately. The default is TRUE (i.e., perform a new rarefaction on the OTU table for each bootstrap sample). However, for cases with large numbers of samples, large numbers of reads per sample or high microbial diversity, this can be slow; thus, there is the option to rarefy the microbiome samples a single time prior to bootstrapping.
seed	Optional seed for random number generator. The default is NULL (i.e., seed picked at random at time of simulation).
dist	How to count taxa present on one, two or three host classes when calculating the 4H Index. Use 'Jaccard' to count each taxon once, regardless of the number of host classes it occurs on (i.e., inspired by the Jaccard Index for beta diversity). Use 'Sorensen' to count each taxon proportional to the number of host classes it occurs on (i.e., inspired by the Sorensen Index for beta diversity). The default is 'Jaccard'.

## Details

FourHbootstrap generates bootstrapped samples of the incidence-based 4H Index. Each bootstrap begins by rarefying the OTU table such that all samples have the same library size. Once the OTU table has been rarefied, a subset, `sample_no`, of hosts is selected from each host class. Hosts can be sampled with or without replacement. The default is to sample hosts without replacement, such that

all hosts in any given bootstrap sample are unique. Subsampled individuals from each host class are then used to determine core microbiomes. This is done by selecting all microbial taxa that occur on a threshold number of individuals of that host class (alternately, different abundance thresholds can also be used, see above). Richness of core microbiomes shared among various host classes are then used to calculate a 4H Index for each bootstrap sample. Specifically, we define:

$$\begin{aligned}
 a &= |(P_1 \cap P_2) \cap H| \\
 b &= |(P_1 \cup P_2) \cap H| - a \\
 b_1 &= |P_1 \cap H| - a \\
 b_2 &= |P_2 \cap H| - a \\
 c &= |H| - a - b \\
 d &= |P_1 \cup P_2 \cup H| - a - b - c \\
 d_1 &= |P_1| - |P_1 \cap P_2| - |P_1 \cap H| + a \\
 d_2 &= |P_2| - |P_1 \cap P_2| - |P_2 \cap H| + a \\
 d_{12} &= d - d_1 - d_2
 \end{aligned}$$

where  $P_1$  is the set of core microbial taxa from the first parent species,  $P_2$  is the set of core microbial taxa from the second parent species,  $H$  is the set of core microbial taxa from hybrid organisms, and  $|S|$  denotes the cardinality of set  $S$ . Thus,  $a$  is the number of microbial taxa shared by both progenitors and the hybrid,  $b$  is the number of microbial taxa shared by one progenitor (but not both) and the hybrid,  $b_1$  is the number of microbial taxa shared by the first progenitor and the hybrid,  $b_2$  is the number of microbial taxa shared by the second progenitor and the hybrid,  $c$  is the number of microbial taxa found only on the hybrid,  $d$  is the number of microbial taxa found only on one or both progenitors,  $d_1$  is the number of microbial taxa found only on the first progenitor,  $d_2$  is the number of microbial taxa found only on the second progenitor, and  $d_{12}$  is the number of microbial taxa found on both progenitors.

When dist='Jaccard', the 4H Index is calculated as:

$$\begin{aligned}
 \mathcal{U} &= \frac{b}{a + b + c + d} \\
 \mathcal{U}_1 &= \frac{b_1}{a + b + c + d} \\
 \mathcal{U}_2 &= \frac{b_2}{a + b + c + d} \\
 \mathcal{I} &= \frac{a}{a + b + c + d} \\
 \mathcal{G} &= \frac{c}{a + b + c + d} \\
 \mathcal{L} &= \frac{d}{a + b + c + d} = 1 - \mathcal{U} - \mathcal{I} - \mathcal{G}
 \end{aligned}$$

$\mathcal{U}$ ,  $\mathcal{I}$ ,  $\mathcal{G}$  and  $\mathcal{L}$  represent the extent of the Union, Intersection, Gain and Loss Models respectively (see Camper et al., 2023) and  $\mathcal{U}_1$  and  $\mathcal{U}_2$  partition the Union model into components associated with the first and second progenitor.

When  $\text{dist} = \text{'Sorensen'}$ , the 4H Index is calculated as:

$$\begin{aligned} \mathcal{U} &= \frac{2b}{3a + 2b + c + d_1 + d_2 + 2d_12} \\ \mathcal{U}_1 &= \frac{2b_1}{3a + 2b + c + d_1 + d_2 + 2d_12} \\ \mathcal{U}_2 &= \frac{2b_2}{3a + 2b + c + d_1 + d_2 + 2d_12} \\ \mathcal{I} &= \frac{3a}{3a + 2b + c + d_1 + d_2 + 2d_12} \\ \mathcal{G} &= \frac{c}{3a + 2b + c + d_1 + d_2 + 2d_12} \\ \mathcal{L} &= \frac{d_1 + d_2 + 2d_12}{3a + 2b + c + d_1 + d_2 + 2d_12} = 1 - \mathcal{U} - \mathcal{I} - \mathcal{G} \end{aligned}$$

In addition to the four dimensions of the 4H Index, FourHbootstrap also calculates  $\sigma$  for each bootstrap sample:

Jaccard:

$$\sigma = \frac{a + d_12}{a + d_12 + d_1 + d_2}$$

Sorensen:

$$\sigma = \frac{2(a + d_12)}{2(a + d_12) + d_1 + d_2}$$

where  $\sigma$  is the fraction of parental microbial taxa that are found on both parent species. This will be used later for calculating the null plane of a given hybrid complex (see FourHnullplane).

## Value

This function returns a matrix, where rows are individual bootstrap samples and columns are the four dimensions of the 4H Index, as well as  $\sigma$ ,  $\mathcal{U}_1$ , and  $\mathcal{U}_2$ .

## References

- Camper, B., Laughlin, Z. et al. "A Conceptual Framework for Host-Associated Microbiomes of Hybrid Organisms" <doi:10.1101/2023.05.01.538925>
- Henry, L., Wickham H., et al. "rlang: Functions for Base Types and Core R and 'Tidyverse' Features" <https://CRAN.R-project.org/package=rclang>
- McMurdie, Paul J., and Susan Holmes. "phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data." *PloS one* 8.4 (2013): e61217.
- McMurdie, Paul J., and Susan Holmes. "Phyloseq: a bioconductor package for handling and analysis of high-throughput phylogenetic sequence data." *Biocomputing* 2012. 2012. 235-246.

**Examples**

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)

FourHbootstrap(enterotype,hybrid_status,0.5,5,10)
```

---

FourHbootstrapA

*Bootstraps of the abundance-based 4H Index*


---

**Description**

Calculates abundance-based 4H Indices for bootstrapped samples drawn from an object of class phyloseq containing host-associated microbial community data from both hybrid hosts and hosts of each parent species. The phyloseq object must, at a minimum, contain an OTU table AND sample data.

**Usage**

```
FourHbootstrapA(x, class_grouping, core_fraction, boot_no, sample_no,
  core_average_abundance = 0, core_max_abundance = 0,
  core_all_average_abundance = 0, core_all_max_abundance = 0,
  replace_hosts=FALSE, reads=NULL, rarefy_each_step=TRUE, seed=NULL,
  dist='Bray-Curtis', representative='mean', rescale_core=FALSE, use_microViz = 'yes')
```

**Arguments**

x	(Required) Host-associated microbial community data . This must be in the form of a phyloseq object and must contain an OTU abundance table with information on the host-associated microbial communities from both hybrid hosts and hosts of each progenitor taxon. The OTU abundance table should contain read counts, not fractional abundances.
class_grouping	(Required) A numerical vector identifying the host class of each host from the OTU table. This vector must be ordered according to the OTU abundance table, and must use the following notation: Progenitor One = 1, Hybrids = 2, Progenitor Two = 3.
core_fraction	(Required) The fraction of hosts that a microbial taxon must be found on to be considered part of a host class's 'core' microbiome
boot_no	(Required) The desired number of bootstrap samples

sample_no	(Required) The number of hosts of each host class that should be selected to generate each bootstrap sample. This number should be less than the minimum number of hosts in any given class.
core_average_abundance	The minimum average abundance at which a microbial taxon must be found across all hosts within a class to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum average abundance)
core_max_abundance	The minimum abundance at which a microbial taxon must be found on at least one host within a class to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum abundance on any single host)
core_all_average_abundance	The minimum average abundance at which a microbial taxon must be found across all hosts (regardless of class) to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum average abundance)
core_all_max_abundance	The minimum abundance at which a microbial taxon must be found on at least one host (regardless of class) to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum abundance on any single host)
replace_hosts	Whether or not to replace hosts during bootstrapping. The default is FALSE (i.e., all hosts in a given bootstrap are unique).
reads	The number of reads to rarefy to for each microbiome sample. The default is to use the number of reads in the sample with the lowest number of reads. Any user-specified value must be lower than the number of reads in the sample with the lowest number of reads.
rarefy_each_step	Whether or not to rarefy each bootstrap sample separately. The default is TRUE (i.e., perform a new rarefaction on the OTU table for each bootstrap sample). However, for cases with large numbers of samples, large numbers of reads per sample or high microbial diversity, this can be slow; thus, there is the option to rarefy the microbiome samples a single time prior to bootstrapping.
seed	Optional seed for random number generator. The default is NULL (i.e., seed picked at random at time of simulation).
dist	How to count taxa present on one, two or three host classes when calculating the 4H Index. Use 'Ruzicka' to count each taxon once, regardless of the number of host classes it occurs on (i.e., inspired by the Ruzicka Index for beta diversity). Use 'Bray-Curtis' to count each taxon proportional to the number of host classes it occurs on (i.e., inspired by the Bray-Curtis Index for beta diversity).
representative	The method used to find representative microbial relative abundances for each host class. Options are 'mean', which is the average abundance of all microbial taxa found on individuals within the host class, and 'median' which is the median abundance of all microbial taxa found on individuals within the host class.

If 'median' is chosen, then representative relative abundances are renormalized for each host class such that total abundances of all microbial taxa (core and non-core) are the same across host classes. The default is 'mean'.

rescale_core	Whether or not to renormalize relative abundances across the subset of microbial taxa that comprise the core of each host class. Renormalizing the core microbial abundances of each host class results in the 4H Index being density invariant. However, this also constrains the 4H Index to a 2D plane, since there is no variation in total abundance across host classes. The default is to use raw relative abundances without renormalizing. Although this means that the 4H Index is not density invariant, in this case, differences in total abundance of the core microbiome between host classes are meaningful since they represent variation in the extent to which the overall microbiome of a host class is shared among individual animals.
use_microViz	Whether or not to use the microViz package to check your phyloseq object. If installation of the microViz package is problematic, this test can be skipped, but the user should verify themselves that a phyloseq object exists AND includes sample data (the sample data placeholder is necessary for the FourHbootstrapA code to work).

## Details

FourHbootstrapA generates bootstrapped samples of the abundance-based 4H Index. Each bootstrap begins by rarefying the OTU table such that all samples have the same library size. Once the OTU table has been rarefied, a subset, `sample_no`, of hosts is selected from each host class. Hosts can be sampled with or without replacement. The default is to sample hosts without replacement, such that all hosts in any given bootstrap sample are unique. Subsampled individuals from each host class are then used to (i) find representative relative abundances of each microbial taxon found on that host class and (ii) identify the core microbiome of the host class. Representative relative abundances are found by averaging or finding the median relative abundance of each microbial taxon across a particular host class. Core microbial taxa are identified by selecting all microbial taxa that occur on a threshold number of individuals of that host class (alternately, different abundance thresholds can also be used, see above). Once representative relative abundances of core microbial taxa have been determined, the core microbiome may or may not be renormalized (the default is to use the core microbial abundances without renormalization). Representative relative abundances are then used to determine whether certain host classes have an excess/deficit of abundance of any particular microbial taxa. Specifically, we define:

$$A = \sum_{i=1}^S \min(\min(x_{iP_1}, x_{iP_2}), x_{iH}) = \sum_{i=1}^S \alpha_i$$

$$B = \sum_{i=1}^S [\min(x_{iP_1} - \alpha_i, x_{iH} - \alpha_i) + \min(x_{iP_2} - \alpha_i, x_{iH} - \alpha_i)] = \sum_{i=1}^S (\beta_{1i} + \beta_{2i})$$

$$B_1 = \sum_{i=1}^S \min(x_{iP_1} - \alpha_i, x_{iH} - \alpha_i) = \sum_{i=1}^S (\beta_{1i})$$

$$B_2 = \sum_{i=1}^S \min(x_{iP_2} - \alpha_i, x_{iH} - \alpha_i) = \sum_{i=1}^S (\beta_{2i})$$



$$C = \sum_{i=1}^S (x_{iH} - \alpha_i - \beta_{1i} - \beta_{2i})$$

$$D = \sum_{i=1}^S [x_{iP_1} + x_{iP_2} - \alpha_i - \beta_{1i} - \beta_{2i} - \min(x_{iP_1}, x_{iP_2})]$$

$$D_1 = \sum_{i=1}^S [x_{iP_1} - \beta_{1i} - \min(x_{iP_1}, x_{iP_2})]$$

$$D_2 = \sum_{i=1}^S [x_{iP_2} - \beta_{2i} - \min(x_{iP_1}, x_{iP_2})]$$

$$D_{12} = \sum_{i=1}^S [\min(x_{iP_1}, x_{iP_2}) - \alpha_i]$$

where  $x_{iP_1}$  is the relative abundance of microbial taxon  $i$  in a representative microbiome of the first parent species,  $x_{iP_2}$  is the relative abundance of microbial taxon  $i$  in a representative microbiome of the second parent species,  $x_{iH}$  is the relative abundance of microbial taxon  $i$  in a representative microbiome of the hybrid organism. Thus,  $A$  is the relative abundance of microbial taxa shared by both progenitors and the hybrid,  $B$  is the relative abundance of microbial taxa shared by one progenitor (but not both) and the hybrid,  $B_1$  is the relative abundance of microbial taxa shared by the first progenitor and the hybrid,  $B_2$  is the relative abundance of microbial taxa shared by the second progenitor and the hybrid,  $C$  is the relative abundance of microbial taxa found only on the hybrid,  $D$  is the relative abundance of microbial taxa found only on one or both progenitors,  $D_1$  is the relative abundance of microbial taxa found only on the first progenitor,  $D_2$  is the relative abundance of microbial taxa found only on the second progenitor, and  $D_{12}$  is the relative abundance of microbial taxa found on both progenitors.

When dist='Ruzicka', the 4H Index is calculated as:

$$U = \frac{B}{A + B + C + D}$$

$$U_1 = \frac{B_1}{A + B + C + D}$$

$$U_2 = \frac{B_2}{A + B + C + D}$$

$$I = \frac{A}{A + B + C + D}$$

$$G = \frac{C}{A + B + C + D}$$

$$L = \frac{D}{A + B + C + D} = 1 - U - I - G$$

$U$ ,  $I$ ,  $G$  and  $L$  represent the extent of the Union, Intersection, Gain and Loss Models respectively (see Camper et al., 2023) and  $U_1$  and  $U_2$  partition the Union model into components associated with the first and second progenitor.

When dist = 'Bray-Curtis', the 4H Index is calculated as:

This allows calculation of the four dimensions of the 4H Index:

$$\begin{aligned} \mathcal{U} &= \frac{2B}{3A + 2B + C + D_1 + D_2 + 2D_{12}} \\ \mathcal{U}_1 &= \frac{2B_1}{3A + 2B + C + D_1 + D_2 + 2D_{12}} \\ \mathcal{U}_2 &= \frac{2B_2}{3A + 2B + C + D_1 + D_2 + 2D_{12}} \\ \mathcal{I} &= \frac{3A}{3A + 2B + C + D_1 + D_2 + 2D_{12}} \\ \mathcal{G} &= \frac{C}{3A + 2B + C + D_1 + D_2 + 2D_{12}} \\ \mathcal{L} &= \frac{D_1 + D_2 + 2D_{12}}{3A + 2B + C + D_1 + D_2 + 2D_{12}} = 1 - \mathcal{U} - \mathcal{I} - \mathcal{G} \end{aligned}$$

In addition to the four dimensions of the 4H Index, FourHbootstrapA also calculates  $\sigma$  for each bootstrap sample:

Ruzicka:

$$\sigma = \frac{A + D_{12}}{A + D_{12} + D_1 + D_2}$$

Bray-Curtis:

$$\sigma = \frac{2(A + D_{12})}{2(A + D_{12}) + D_1 + D_2}$$

where  $\sigma$  is the fraction of parental microbial taxa that are found on both parent species. This will be used later for calculating the null plane of a given hybrid complex (see FourHnullplane).

## Value

This function returns a matrix, where rows are individual bootstrap samples and columns are the four dimensions of the 4H Index, as well as  $\sigma$ ,  $\mathcal{U}_1$ , and  $\mathcal{U}_2$ .

## References

- Camper, B., Laughlin, Z. et al. "A Conceptual Framework for Host-Associated Microbiomes of Hybrid Organisms" <doi:10.1101/2023.05.01.538925>
- Henry, L., Wickham H., et al. "rlang: Functions for Base Types and Core R and 'Tidyverse' Features" <https://CRAN.R-project.org/package=rlang>
- McMurdie, Paul J., and Susan Holmes. "phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data." PloS one 8.4 (2013): e61217.
- McMurdie, Paul J., and Susan Holmes. "Phyloseq: a bioconductor package for handling and analysis of high-throughput phylogenetic sequence data." Bioinformatics 2012. 2012. 235-246.

**Examples**

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)

FourHbootstrapA(enterotype,hybrid_status,0.5,5,10, use_microViz='no')
```

---

FourHcentroid

*Centroid of bootstrapped 4H-indices*


---

**Description**

Calculates the centroid of a set of 4H-indices from bootstrapped samples.

**Usage**

```
FourHcentroid(boots)
```

**Arguments**

boots (Required) Output matrix from the FourHbootstrap function

**Details**

FourHcentroid averages over each column of the output matrix from FourHbootstrap and prints out the average value of each dimension ( $\mathcal{U}$ ,  $\mathcal{I}$ ,  $\mathcal{G}$ ,  $\mathcal{L}$ ) of the 4H-index. It also calculates and prints out the average value for  $\sigma$ .

**Value**

This function outputs a dataframe reporting the average values of each dimension of the 4H-index, and the average value of  $\sigma$ .

**Examples**

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
```

```
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)

#Bootstrap the dataset
boot_samples<-FourHbootstrap(enterotype,hybrid_status,0.5,5,10)

#Find the centroid of the bootstrap samples
FourHcentroid(boot_samples)
```

---

FourHcompare

*Tests whether multiple systems have significantly different 4H-indices*


---

### Description

FourHcompare uses a PERMANOVA test on the log-ratio transformed bootstrap samples of the 4H-index from multiple different hybrid systems to determine whether the systems are significantly different in terms of the relative importance of the Union, Intersection, Gain and Loss Models.

### Usage

```
FourHcompare(boots_sets, boots_types, method='ilr', permutations = 1000)
```

### Arguments

boots_sets	(Required) A matrix of bootstrapped 4H-indices from all of the systems that are to be compared. Rows are individual bootstraps, and columns are the four dimensions of the 4H-index and sigma from a particular bootstrap sample. The simplest way to generate this matrix is to use <code>rbind()</code> to merge the outputs of <code>FourHbootstrap</code> from multiple different systems.
boots_types	(Required) A vector specifying which hybrid system each bootstrap sample belongs to. The order of this vector must be the same as the order of the bootstrapped samples in the matrix of bootstrapped 4H indices.
method	The method used to transform the compositional 4H-index in the 4-part Aitchison Simplex to a 3-dimensional euclidean vector. Statistical tests are performed on the 3-dimensional euclidean vectors. Currently, options for transformation are 'ilr' for an isometric log-ratio transformation, 'clr' for a centered log-ratio transformation, 'alr' for an additive log-ratio transformation and 'none' for non-transformed data. The default is to use an isometric log-ratio transformation.
permutations	The desired number of permutations to use for the PERMANOVA. The default is 1000.

### Details

FourHcompare takes bootstrapped samples of the 4H-index from multiple different hybrid systems. First, each 4H-index from each bootstrapped sample is transformed to a 3-dimensional euclidean vector using a log-ratio transformation. This is done using the `ilr` or `clr` functions from the `compositions` R package. Next, the PERMANOVA function from the PERMANOVA R package is used to test whether there are significant differences between the 4H indices of the different hybrid systems.

**Value**

This function creates a list with contents inherited from the PERMANOVA function.

**References**

Van den Boogaart, K. Gerald, and Raimon Tolosana-Delgado. "“Compositions”: a unified R package to analyze compositional data." *Computers & Geosciences* 34.4 (2008): 320-338.

van den Boogaart, K. Gerald, et al. "Package ‘compositions’." *Compositional data analysis Ver* (2013): 1-40.

van den Boogaart, K. Gerald, Tolosana-Delgado, R., Bren, M. "compositions: Compositional Data Analysis" <https://CRAN.R-project.org/package=compositions>

Vicente-Gonzalez, L., J.L. Vicente-Villardón. "PERMANOVA: Multivariate Analysis of Variance Based on Distances and Permutations" <https://CRAN.R-project.org/package=PERMANOVA>

Henry, L., Wickham H., et al. "rlang: Functions for Base Types and Core R and ‘Tidyverse’ Features" <https://CRAN.R-project.org/package=rlang>

**Examples**

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Use a subset of the enterotype dataset for one hybrid system...
#...and a subset for a second hybrid system
TS<-prune_samples(grepl('TS',sample_names(enterotype)),enterotype)
MH<-prune_samples(grepl('MH',sample_names(enterotype)),enterotype)

#Randomly assign host classes (these should be known in real hybrid microbiome datasets)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_statusTS<-sample(1:3,154, replace=TRUE)
hybrid_statusMH<-sample(1:3,85, replace=TRUE)

#Calculate bootstrapped samples of the 4H-index for each of the two systems
bootstrapTS<-FourHbootstrap(TS,hybrid_statusTS,0.5,5,10)
bootstrapMH<-FourHbootstrap(MH,hybrid_statusMH,0.5,5,10)

#Bind the two system 4H-index matrices together
boots_to_compare<-rbind(bootstrapTS,bootstrapMH)
system_info<-c(rep(1,5),rep(2,5))

#Perform the PERMANOVA
FourHcompare(boots_to_compare, system_info, method='ilr', permutations = 1000)
```

FourHnull

*Bootstrapped null models of the 4H-index***Description**

Calculates 4H-indices for bootstrapped samples of hybrid null models drawn from an object of class phyloseq containing host-associated microbial community data from both hybrid hosts and hosts of each parent species.

**Usage**

```
FourHnull(x, class_grouping, core_fraction, boot_no,
          sample_no, null_model=1, replace_hosts=FALSE, reads=NULL,
          rarefy_each_step=TRUE, seed=NULL, use_microViz='yes')
```

**Arguments**

x	(Required) Host-associated microbial community data . This must be in the form of a phyloseq object and must contain an OTU abundance table with information on the host-associated microbial communities from both hybrid hosts and hosts of each progenitor taxon. The OTU abundance table should contain read counts, not fractional abundances.
class_grouping	(Required) A numerical vector identifying the host class of each host from the OTU table. This vector must be ordered according to the OTU abundance table, and must use the following notation: Progenitor One = 1, Hybrids = 2, Progenitor Two = 3.
core_fraction	(Required) The fraction of hosts that a microbial taxon must be found on to be considered part of a host class's 'core' microbiome
boot_no	(Required) The desired number of bootstrap samples
sample_no	(Required) The number of hosts of each host class that should be selected to generate each bootstrap sample. This number should be less than the minimum number of hosts in any given class.
null_model	The particular null model to use (see below for a description of each null model). The default is 1 (averaging microbial taxon abundances from a randomly drawn individual of each progenitor species)
replace_hosts	Whether or not to replace hosts during bootstrapping. The default is FALSE (i.e., all hosts in a given bootstrap are unique).
reads	The number of reads to rarefy to for each microbiome sample. The default is to use the number of reads in the sample with the lowest number of reads. Any user-specified value must be lower than the number of reads in the sample with the lowest number of reads.
rarefy_each_step	Whether or not to rarefy each bootstrap sample separately. The default is TRUE (i.e., perform a new rarefaction on the OTU table for each bootstrap sample). However, for cases with large numbers of samples, large numbers of reads per

	sample or high microbial diversity, this can be slow; thus, there is the option to rarefy the microbiome samples a single time prior to bootstrapping.
seed	Optional seed for random number generator. The default is NULL (i.e., seed picked at random at time of simulation).
use_microViz	Whether or not to use the microViz package to check your phyloseq object. If installation of the microViz package is problematic, this test can be skipped, but the user should verify themselves that a phyloseq object exists AND includes sample data (the sample data placeholder is necessary for the FourHnull code to work).

## Details

FourHnull generates bootstrapped samples of hybrid null models of the 4H index. As in FourHbootstrap, each iteration begins by rarefying the OTU table such that all samples have the same library size. Once the OTU table has been rarefied, a subset, `sample_no`, of hosts is selected from each progenitor class. Hosts can be sampled with or without replacement. The default is to sample hosts without replacement, such that all progenitors in any given bootstrap sample are unique. Next, `sample_no` hybrid individuals are generated according to the chosen null model. Null models are as follows:

`null_model= 1`: each hybrid individual is created by randomly selecting one individual from the progenitor 1 class and one individual from the progenitor 2 class and then averaging the microbial abundances from the two progenitor individuals.

`null_model= 2`: each hybrid individual is created by randomly selecting two individuals from the progenitor 1 class and one individual from the progenitor 2 class and then averaging the microbial abundances from the three progenitor individuals.

`null_model= 3`: each hybrid individual is created by randomly selecting one individual from the progenitor 1 class and two individuals from the progenitor 2 class and then averaging the microbial abundances from the three progenitor individuals.

`null_model= 4`: each hybrid individual is created by randomly selecting two individual from the progenitor 1 class and then averaging the microbial abundances from these two progenitor individuals.

`null_model= 5`: each hybrid individual is created by randomly selecting two individual from the progenitor 2 class and then averaging the microbial abundances from these two progenitor individuals.

`null_model= 6`: each hybrid individual is created by randomly selecting one individual from the progenitor 1 class and using its microbial abundances.

`null_model= 7`: each hybrid individual is created by randomly selecting one individual from the progenitor 2 class and using its microbial abundances.

`null_model= 10`: each hybrid individual is created by randomly selecting one individual from the progenitor 1 class and one individual from the progenitor 2 class, randomly setting half of the microbial taxon abundances only found on the progenitor 1 individual to zero, randomly setting half of the microbial taxon abundances only found on the progenitor 2 individual to zero, and then setting all other microbial taxon abundances to one.

Once null model hybrid iterates have been generated, FourHnull proceeds as in FourHbootstrap and calculates the 4H-index and  $\sigma$  for each bootstrap sample.

**Value**

This function returns a matrix, where rows are individual bootstrap samples and columns are the four dimensions of the 4H-index, as well as  $\sigma$ .

**References**

Camper, B., Laughlin, Z. et al. "A Conceptual Framework for Host-Associated Microbiomes of Hybrid Organisms" <doi:10.1101/2023.05.01.538925>

Henry, L., Wickham H., et al. "rlang: Functions for Base Types and Core R and 'Tidyverse' Features" <https://CRAN.R-project.org/package=rclang>

McMurdie, Paul J., and Susan Holmes. "phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data." PloS one 8.4 (2013): e61217.

McMurdie, Paul J., and Susan Holmes. "Phyloseq: a bioconductor package for handling and analysis of high-throughput phylogenetic sequence data." Biocomputing 2012. 2012. 235-246.

**Examples**

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)

FourHnull(enterotype,hybrid_status,0.5,5,10, use_microViz='no')
```

---

FourHnullplane

*Null Plane*

---

**Description**

Calculates and draws the null plane on a quaternary (four dimensional barycentric, Aitchison Simplex) plot.

**Usage**

```
FourHnullplane(boots,col='red')
```

**Arguments**

**boots** (Required) Output matrix from the FourHbootstrap function

**col** The color of the plane in the quaternary plot. The default color is red.



## Details

FourHnullplane uses the average value of  $\sigma = (P_1 \cap P_2)/(P_1 \cup P_2)$  across all bootstrap samples to draw a null plane on an existing quaternary plot. For any given value of  $\theta = \mathcal{G} + \mathcal{L}$ , the null plane is defined as follows:

$$\mathcal{U}_{null} = (1 - \sigma)(1 - \theta)$$

$$\mathcal{I}_{null} = \sigma(1 - \theta)$$

The primary purpose of the null plane is visualization. In particular, the null plane reflects the ratio of  $(I)/(\mathcal{U} + \mathcal{I})$  that would be expected if hybrids were equally likely to acquire any microbial taxa found on one or both parent species. 4H-indices that lie closer to the  $\mathcal{I}$  vertex than the null plane indicate that hybrid organisms are more likely to retain microbial taxa shared by both parent species than microbial taxa found on only one parent species (and by corollary, hybrid organisms are more likely to lose microbial taxa found on only one parent species than they are to lose microbial taxa found on both parent species). 4H-indices that lie closer to the  $\mathcal{U}$  vertex than the null plane indicate that hybrid organisms are more likely to retain microbial taxa found on only one parent species than microbial taxa found on both parent species (and by corollary, hybrid organisms are more likely to lose microbial taxa found on both parent species than they are to lose microbial taxa found on only one parent species).

## Value

This function outputs a plane on an existing quaternary plot.

## References

- Henry, L., Wickham H., et al. "rlang: Functions for Base Types and Core R and 'Tidyverse' Features" <https://CRAN.R-project.org/package=rlang>
- Van den Boogaart, K. Gerald, and Raimon Tolosana-Delgado. "'Compositions': a unified R package to analyze compositional data." *Computers & Geosciences* 34.4 (2008): 320-338.
- van den Boogaart, K. Gerald, et al. "Package 'compositions'." *Compositional data analysis* Ver (2013): 1-40.
- van den Boogaart, K. Gerald, Tolosana-Delgado, R., Bren, M. "compositions: Compositional Data Analysis" <https://CRAN.R-project.org/package=compositions>
- Adler, Daniel, Oleg Nenadic, and Walter Zucchini. "Rgl: A r-library for 3d visualization with opengl." *Proceedings of the 35th symposium of the interface: computing science and statistics*, Salt Lake City. Vol. 35. 2003.
- Adler, D., Murdoch, M. D., Suggests, M. A. S. S., WebGL, P. L. Y., OBJ, S., & OpenGL, S. (2019). Package 'rgl'.
- Murdoch, Duncan, Daniel Adler, and Oleg Nenadic. "Package 'rgl'." *R Package* (2023).

## Examples

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))
```

```

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)

#Bootstrap the dataset
boot_samples<-FourHbootstrap(enterotype,hybrid_status, 0.5,5,10)

#Plot the null plane on an existing quaternary plot
#An existing quaternary plot should already be open
FourHnullplane(boot_samples,col='red')

```

---

FourHnullplaneD

*Bootstrap Distribution Around the Null Plane*


---

## Description

Calculates the distribution of bootstrap samples around the null expectation for Union vs. Intersection models.

## Usage

```
FourHnullplaneD(boots)
```

## Arguments

boots (Required) Output matrix from the FourHbootstrap function

## Details

For each bootstrap sample, FourHnullplaneD uses the fraction of microbes shared by both progenitors to calculate null model expectations for the value of  $\mathcal{U}$  and  $\mathcal{I}$ . It then calculates the difference between the observed and expected values along the intersection and union dimensions  $\mathcal{I} - \mathcal{I}_{null}$  and  $\mathcal{U} - \mathcal{U}_{null}$ . Finally, FourHnullplaneD plots a histogram of the difference between observed and expected intersection dimensions,  $\mathcal{I} - \mathcal{I}_{null}$  and calculates the average distance between the observed and expected intersection dimensions, as well as the standard deviation of  $\mathcal{I} - \mathcal{I}_{null}$  and the fraction of bootstrapped samples where  $\mathcal{I} - \mathcal{I}_{null} < 0$  (i.e., hybrids preferentially retain microbial taxa only found on one progenitor). Preferential retention of microbial taxa found on both progenitors can then be assessed as >95% of bootstrapped samples having  $\mathcal{I} - \mathcal{I}_{null} > 0$ .

## Value

This function outputs a list including a dataframe with the following lists:

sigma The inputted values of  $\sigma$  for each bootstrapped sample.  
theta The calculated values of  $\theta = \mathcal{G} + \mathcal{L}$  for each bootstrapped sample.

union	The observed values of the union dimension, $\mathcal{U}$ , for each bootstrapped sample.
union_expectation	The expected values of the union dimension, $\mathcal{U}_{null}$ , for each bootstrapped sample.
intersection	The observed values of the intersection dimension, $\mathcal{I}$ , for each bootstrapped sample.
intersection_expectation	The expected values of the intersection dimension, $\mathcal{I}_{null}$ , for each bootstrapped sample.
diffI	The difference between the observed and expected values of the intersection dimension for each bootstrapped sample.
diffU	The difference between the observed and expected values of the union dimension for each bootstrapped sample.

as well as the average distance and standard deviation between the observed and expected intersection dimensions, and the fraction of bootstrapped samples where hybrids preferentially retain microbial taxa only found on one progenitor.

### Examples

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)

#Bootstrap the dataset
boot_samples<-FourHbootstrap(enterotype,hybrid_status,0.5,5,10)

#Calculate the expected and observed distributions of bootstrapped samples around the null plane
FourHnullplaneD(boot_samples)
```

---

FourHpreanalysis

*Bootstraps of the percentage of gamma diversity that is core*

---

### Description

Calculates the percentage of the total gamma diversity that is part of the core microbiota for bootstrapped samples drawn from an object of class phyloseq containing host-associated microbial community data from both hybrid hosts and hosts of each parent species. Prints out a warning when the percentage of the total gamma diversity that is core for hybrid hosts is less than half of the average percentage that is core for the two parent populations.

**Usage**

```
FourHpreanalysis(x, class_grouping, core_fraction, boot_no, sample_no,
  core_average_abundance = 0, core_max_abundance = 0,
  core_all_average_abundance = 0, core_all_max_abundance = 0,
  replace_hosts=FALSE, reads=NULL, rarefy_each_step=TRUE, seed=NULL)
```

**Arguments**

- x** (Required) Host-associated microbial community data . This must be in the form of a phyloseq object and must contain an OTU abundance table with information on the host-associated microbial communities from both hybrid hosts and hosts of each progenitor taxon. The OTU abundance table should contain read counts, not fractional abundances.
- class\_grouping** (Required) A numerical vector identifying the host class of each host from the OTU table. This vector must be ordered according to the OTU abundance table, and must use the following notation: Progenitor One = 1, Hybrids = 2, Progenitor Two = 3.
- core\_fraction** (Required) The fraction of hosts that a microbial taxon must be found on to be considered part of a host class's 'core' microbiome
- boot\_no** (Required) The desired number of bootstrap samples
- sample\_no** (Required) The number of hosts of each host class that should be selected to generate each bootstrap sample. This number should be less than the minimum number of hosts in any given class.
- core\_average\_abundance**  
The minimum average abundance at which a microbial taxon must be found across all hosts within a class to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum average abundance)
- core\_max\_abundance**  
The minimum abundance at which a microbial taxon must be found on at least one host within a class to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum abundance on any single host)
- core\_all\_average\_abundance**  
The minimum average abundance at which a microbial taxon must be found across all hosts (regardless of class) to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum average abundance)
- core\_all\_max\_abundance**  
The minimum abundance at which a microbial taxon must be found on at least one host (regardless of class) to be considered part of a host class's 'core' microbiome. The default is zero (i.e., it does not need to occur at any minimum abundance on any single host)
- replace\_hosts** Whether or not to replace hosts during bootstrapping. The default is FALSE (i.e., all hosts in a given bootstrap are unique).

reads	The number of reads to rarefy to for each microbiome sample. The default is to use the number of reads in the sample with the lowest number of reads. Any user-specified value must be lower than the number of reads in the sample with the lowest number of reads.
rarefy_each_step	Whether or not to rarefy each bootstrap sample separately. The default is TRUE (i.e., perform a new rarefaction on the OTU table for each bootstrap sample). However, for cases with large numbers of samples, large numbers of reads per sample or high microbial diversity, this can be slow; thus, there is the option to rarefy the microbiota samples a single time prior to bootstrapping.
seed	Optional seed for random number generator. The default is NULL (i.e., seed picked at random at time of simulation).

## Details

FourHpreanalysis generates bootstrapped samples of the percentage of gamma diversity for hybrids and each parent species that is part of the core. Each bootstrap begins by rarefying the OTU table such that all samples have the same library size. Once the OTU table has been rarefied, a subset, `sample_no`, of hosts is selected from each host class. Hosts can be sampled with or without replacement. The default is to sample hosts without replacement, such that all hosts in any given bootstrap sample are unique. Subsampled individuals from each host class are then used to determine overall gamma diversity and diversity of core microbiotas for hybrid hosts and hosts of each parent species. This allows calculation of the percentage of gamma diversity that is part of the core. When this percentage among hybrids is less than half of what it is, on average, among parent species, a warning is printed.

## References

Camper, B., Laughlin, Z. et al. "A Conceptual Framework for Host-Associated Microbiomes of Hybrid Organisms" <doi:10.1101/2023.05.01.538925>

Henry, L., Wickham H., et al. "rlang: Functions for Base Types and Core R and 'Tidyverse' Features" <https://CRAN.R-project.org/package=rlang>

McMurdie, Paul J., and Susan Holmes. "phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data." *PloS one* 8.4 (2013): e61217.

McMurdie, Paul J., and Susan Holmes. "Phyloseq: a bioconductor package for handling and analysis of high-throughput phylogenetic sequence data." *Bioinformatics* 2012. 2012. 235-246.

## Examples

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)
```

```
FourHpreanalysis(enterotype,hybrid_status,0.5,5,10)
```

---

FourHquaternary      *Plot bootstrapped 4H-indices*

---

### Description

Draws bootstrapped values of the 4H-index on a quaternary (four dimensional barycentric, Aitchison Simplex) plot with an additional option to draw the centroid of the bootstrapped samples as well.

### Usage

```
FourHquaternary(boots,col='red',addplot=FALSE,plotcentroid=TRUE,
                plotgrid=TRUE, size_bootstrap=1,size_centroid=15,size_font=2)
```

### Arguments

boots	(Required) Output matrix from the FourHbootstrap function
col	The color of the points in the quaternary plot. The default is RED
addplot	Whether or not to add the data to an existing plot. The default is FALSE
plotcentroid	Whether or not to plot the centroid of the bootstrap samples. The default is TRUE.
plotgrid	Whether or not to plot gridlines on the quaternary plot. The default is TRUE
size_bootstrap	The size of each point representing a single bootstrap sample. The default is 1.
size_centroid	The size of the point representing the centroid of the bootstrap samples. The default is 15.
size_font	The size of the font used to label the four vertices. The default is 2.

### Details

FourHquaternary allows for visualization of bootstrapped 4H-indices within the framework of a 4-dimensional Aitchison Simplex, also known as a 4-dimensional barycentric plot or quaternary plot. Commonly used for compositional data, the Aitchison Simplex is convenient for the 4H-index, because  $\mathcal{U} + \mathcal{I} + \mathcal{G} + \mathcal{L} = 1$ , reflecting the fact that each of the four different conceptual models - Union, Intersection, Gain and Loss - comprise one part of the whole, where the whole is the total microbial diversity across the entire hybrid host complex. FourHquaternary uses functions from the compositions R package to transform the four coordinates of the 4H-index, such that they can be drawn on a quaternary plot.

### Value

This function plots bootstrapped 4H-indices on quaternary plot (Aitchison Simplex, 4-dimensional barycentric plot).

## References

- Henry, L., Wickham H., et al. "rlang: Functions for Base Types and Core R and 'Tidyverse' Features" <https://CRAN.R-project.org/package=rclang>
- Van den Boogaart, K. Gerald, and Raimon Tolosana-Delgado. "'Compositions': a unified R package to analyze compositional data." *Computers & Geosciences* 34.4 (2008): 320-338.
- van den Boogaart, K. Gerald, et al. "Package 'compositions'." *Compositional data analysis Ver* (2013): 1-40.
- van den Boogaart, K. Gerald, Tolosana-Delgado, R., Bren, M. "compositions: Compositional Data Analysis" <https://CRAN.R-project.org/package=compositions>
- Adler, Daniel, Oleg Nenadic, and Walter Zucchini. "Rgl: A r-library for 3d visualization with opengl." *Proceedings of the 35th symposium of the interface: computing science and statistics, Salt Lake City. Vol. 35. 2003.*
- Adler, D., Murdoch, M. D., Suggests, M. A. S. S., WebGL, P. L. Y., OBJ, S., & OpenGL, S. (2019). Package 'rgl'.
- Murdoch, Duncan, Daniel Adler, and Oleg Nenadic. "Package 'rgl'." *R Package* (2023).

## Examples

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)

#Bootstrap the dataset
boot_samples<-FourHbootstrap(enterotype,hybrid_status,0.5,5,10)

#Plot the bootstrapped samples as a new quaternary plot including the centroid
FourHquaternary(boot_samples,col='red')

#Plot the bootstrapped samples on an existing quaternary plot without including the centroid
#An existing quaternary plot should already be open
FourHquaternary(boot_samples,col='red',addplot=TRUE, plotcentroid=FALSE)
```

---

FourHquaternaryC

*Plot centroid of bootstrapped 4H-indices*

---

## Description

Draws the centroid of bootstrapped values of the 4H-index on a quaternary (four dimensional barycentric, Aitchison Simplex) plot.

**Usage**

```
FourHquaternaryC(boots,col='red',addplot=FALSE,plotgrid=TRUE,
                 size_centroid=15,size_font=2)
```

**Arguments**

boots	Output from the FourHbootstrap function
col	The color of the points in the quaternary plot. The default is 'red'
addplot	Whether or not to add the data to an existing plot. The default is FALSE
plotgrid	Whether or not to plot gridlines on the quaternary plot. The default is TRUE
size_centroid	The size of the point representing the centroid of the bootstrap samples. The default is 15.
size_font	The size of the font used to label the four vertices. The default is 2.

**Details**

FourHquaternaryC allows for visualization of the centroid of bootstrapped 4H-indices within the framework of a 4-dimensional Aitchison Simplex, also known as a 4-dimensional barycentric plot or quaternary plot. Commonly used for compositional data, the Aitchison Simplex is convenient for the 4H-index, because  $\mathcal{U} + \mathcal{I} + \mathcal{G} + \mathcal{L} = 1$ , reflecting the fact that each of the four different conceptual models - Union, Intersection, Gain and Loss - comprise one part of the whole, where the whole is the total microbial diversity across the entire hybrid host complex. FourHquaternaryC uses functions from the `compositions` R package to transform the four coordinates of the 4H-index, such that they can be drawn on a quaternary plot.

**Value**

This function plots the centroid of bootstrapped 4H-indices on quaternary plot (Aitchison Simplex, 4-dimensional barycentric plot).

**References**

- Henry, L., Wickham H., et al. "rlang: Functions for Base Types and Core R and 'Tidyverse' Features" <https://CRAN.R-project.org/package=rlang>
- Van den Boogaart, K. Gerald, and Raimon Tolosana-Delgado. "'Compositions': a unified R package to analyze compositional data." *Computers & Geosciences* 34.4 (2008): 320-338.
- van den Boogaart, K. Gerald, et al. "Package 'compositions'." *Compositional data analysis* Ver (2013): 1-40.
- van den Boogaart, K. Gerald, Tolosana-Delgado, R., Bren, M. "compositions: Compositional Data Analysis" <https://CRAN.R-project.org/package=compositions>
- Adler, Daniel, Oleg Nenadic, and Walter Zucchini. "Rgl: A r-library for 3d visualization with opengl." *Proceedings of the 35th symposium of the interface: computing science and statistics*, Salt Lake City. Vol. 35. 2003.
- Adler, D., Murdoch, M. D., Suggests, M. A. S. S., WebGL, P. L. Y., OBJ, S., & OpenGL, S. (2019). Package 'rgl'.
- Murdoch, Duncan, Daniel Adler, and Oleg Nenadic. "Package 'rgl'." R Package (2023).



**Examples**

```
#Test with enterotype dataset
library(phyloseq)
data(enterotype)
#Covert the OTU table to reads, rather than fractional abundances
otu_table(enterotype)<-round(10000*otu_table(enterotype))

#Randomly assign host classes (these should be known in a real hybrid microbiome dataset)
#The two parent species are assigned '1' and '3' respectively, the hybrid is assigned '2'
hybrid_status<-sample(1:3,280, replace=TRUE)

#Bootstrap the dataset
boot_samples<-FourHbootstrap(enterotype,hybrid_status,0.5,5,10)

#Plot the centroid of the bootstrapped samples as a new quaternary plot
FourHquaternaryC(boot_samples,col='red')

#Plot the centroid of the bootstrapped samples on an existing quaternary plot
#An existing quaternary plot should already be open
FourHquaternaryC(boot_samples,col='red',addplot=TRUE)
```

# Index

FourHbootstrap, [2](#)  
FourHbootstrapA, [6](#)  
FourHcentroid, [11](#)  
FourHcompare, [12](#)  
FourHnull, [14](#)  
FourHnullplane, [16](#)  
FourHnullplaneD, [18](#)  
FourHpreanalysis, [19](#)  
FourHquaternary, [22](#)  
FourHquaternaryC, [23](#)