Package ‘HMP’

October 12, 2022

Type Package

Title Hypothesis Testing and Power Calculations for Comparing Metagenomic Samples from HMP

Version 2.0.1

Date 2019-08-28

Author Patricio S. La Rosa, Elena Deych, Sharina Carter, Berkley Shands, Dake Yang, William D. Shannon

Maintainer Berkley Shands <rpackages@biorankings.com>

Depends R (>= 3.1.0), dirmult

Imports ggplot2, stats, foreach, doParallel, MASS, vegan, gplots, rpart, rpart.plot, parallel, graphics, lattice

Description Using Dirichlet-Multinomial distribution to provide several functions for formal hypothesis testing, power and sample size calculations for human microbiome experiments.

License Apache License (== 2.0)

LazyData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-31 11:00:06 UTC

R topics documented:

`HMP-package` *Hypothesis Testing and Power Calculations for Comparing Metagenomics Samples*

Description

This package provides tools for Generating data matrices following Multinomial and Dirichlet-Multinomial distributions, Computing the following test-statistics and their corresponding p-values, and Computing the power and size of the tests described above using Monte-Carlo simulations.

Details

Hypothesis Test	**Test Statistics Function**	**Power Calculation Function**
2+ Sample Means w/ Reference Vector	`Xmc.sevsample`	`MC.Xmc.statistics`
1 Sample Mean w/ Reference Vector	`Xsc.onesample`	`MC.Xsc.statistics`
2+ Sample Means w/o Reference Vector	`Xmcupo.sevsample`	`MC.Xmcupo.statistics`
2+ Sample Overdispersions	`Xoc.sevsample`	`MC.Xoc.statistics`
2+ Sample DM-Distribution	`Xdc.sevsample`	`MC.Xdc.statistics`
Multinomial vs DM	`C.alpha.multinomial`	`MC.ZT.statistics`

In addition to hypothesis testing and power calculations you can:

1. Perform basic data management to exclude samples with fewer than pre-specified number of reads, collapse rare taxa and order the taxa by frequency. This is useful to exclude failed samples (i.e. samples with very few reads) - `Data.filter`
2. Plot your data - `Barchart.data`
3. Generate random sample of Dirichlet Multinomial data with pre-specified parameters - `Dirichlet.multinomial`

Note: Thought the description of the functions refer its application to ranked abundance distributions (RAD) data, every function is also applicable to model species abundance data. See references for a discussion and application to both type of ecological data.

Author(s)

Patricio S. La Rosa, Elena Deych, Berkley Shands, Sharina Carter, Dake Yang, William D. Shannon

References

La Rosa PS, Brooks JP, Deych E, Boone EL, Edwards DJ, et al. (2012) Hypothesis Testing and Power Calculations for Taxonomic-Based Human Microbiome Data. PLoS ONE 7(12): e52078. doi:10.1371/journal.pone.0052078

Yang D, Johnson J, Zhou X, Deych E, et al. (2019) Microbiome Recursive Partitioning. Currently Under Review.

`Barchart.data` *A Graphical Representation of Taxa Proportions*

Description

Creates a bar plot of taxonomic proportions.

Usage

```
Barchart.data(data, title = "Taxa Proportions")
```

Arguments

`data`	A matrix of taxonomic counts(columns) for each sample(rows).
`title`	A string to be used as the plots title. The default is "Taxa Proportions".

Value

A bar plot of taxonomic proportions for all samples at a given taxonomic level.

Examples

```
data(saliva)

Barchart.data(saliva)
```

C.alpha.multinomial *$C(\alpha)$ - Optimal Test for Assessing Multinomial Goodness of Fit Versus Dirichlet-Multinomial Alternative*

Description

A function to compute the $C(\alpha)$-optimal test statistics of Kim and Margolin (1992) for evaluating the Goodness-of-Fit of a Multinomial distribution (null hypothesis) versus a Dirichlet-Multinomial distribution (alternative hypothesis).

Usage

```
C.alpha.multinomial(data)
```

Arguments

`data`	A matrix of taxonomic counts(columns) for each sample(rows).

Details

In order to test if a set of ranked-abundance distribution(RAD) from microbiome samples can be modeled better using a multinomial or Dirichlet-Multinomial distribution, we test the hypothesis $\mathrm{H} : \theta = 0$ versus $\mathrm{H} : \theta \neq 0$, where the null hypothesis implies a multinomial distribution and the alternative hypothesis implies a DM distribution. Kim and Margolin (Kim and Margolin, 1992) proposed a $C(\alpha)$-optimal test- statistics given by,

$$T = \sum_{j=1}^{K} \sum_{i=1}^{P} \frac{1}{\sum_{i=1}^{P} x_{ij}} \left(x_{ij} - \frac{N_i \sum_{i=1}^{P} x_{ij}}{N_{\mathrm{g}}} \right)^2$$

Where K is the number of taxa, P is the number of samples, x_{ij} is the taxon j, $j = 1, \ldots, K$ from sample i, $i = 1, \ldots, P$, N_i is the number of reads in sample i, and N_{g} is the total number of reads across samples.

As the number of reads increases, the distribution of the T statistic converges to a Chi-square with degrees of freedom equal to $(P - 1)(K - 1)$, when the number of sequence reads is the same in all samples. When the number of reads is not the same in all samples, the distribution becomes a weighted Chi-square with a modified degree of freedom (see (Kim and Margolin, 1992) for more details).

Note: Each taxa in `data` should be present in at least 1 sample, a column with all 0's may result in errors and/or invalid results.

Value

A list containing the $C(\alpha)$-optimal test statistic and p-value.

References

Kim, B. S., and Margolin, B. H. (1992). Testing Goodness of Fit of a Multinomial Model Against Overdispersed Alternatives. Biometrics 48, 711-719.

Examples

```
data(saliva)

calpha <- C.alpha.multinomial(saliva)
calpha
```

Data.filter *A Data Filter*

Description

This function creates a new dataset from an existing one by ordering taxa in order of decreasing abundance, collapsing less-abundant taxa into one category as specified by the user and excluding samples with a total number of reads fewer than the user-specified value.

Usage

```
Data.filter(data, order.type = "data", minReads = 0, numTaxa = NULL,
perTaxa = NULL)
```

Arguments

`data`	A matrix of taxonomic counts(columns) for each sample(rows).
`order.type`	If `"sample"`: Rank taxa based on its taxonomic frequency. If `"data"`: Rank taxa based on cumulative taxonomic counts across all samples (default).
`minReads`	Samples with a total number of reads less than read.crit value will be deleted.
`numTaxa`	The number of taxa to keep, while collapsing the other (less abundant) taxa. Only one argument, numTaxa or perTaxa should be specified.
`perTaxa`	The combined percentage of data to keep, while collapsing the remaining taxa. Only one argument, numTaxa or perTaxa should be specified.

Value

A data frame of taxa and samples with a total number of reads greater than the minimum value. The last taxon labeled 'Other' contains the sum of the least abundant taxa collapsed by setting 'numTaxa' or 'perTaxa'.

Examples

```
data(saliva)

### Excludes all samples with fewer than 1000 reads and collapses
### taxa with 11th or smaller abundance into one category
filterDataNum <- Data.filter(saliva, "data", 1000, numTaxa=10)

### Excludes all samples with fewer than 1000 reads and collapses
### the least abundant taxa to keep as close to 95% of the data as
### possible
filterDataPer <- Data.filter(saliva, "data", 1000, perTaxa=.95)

dim(saliva)
dim(filterDataNum)
dim(filterDataPer)
```

`Dirichlet.multinomial` *Generation of Dirichlet-Multinomial Random Samples*

Description

Random generation of Dirichlet-Multinomial samples.

Usage

```
Dirichlet.multinomial(Nrs, shape)
```

Arguments

`Nrs`	A vector specifying the number of reads or sequence depth for each sample.
`shape`	A vector of Dirichlet parameters for each taxa.

Details

The Dirichlet-Multinomial distribution is given by (Mosimann, J. E. (1962); Tvedebrink, T. (2010)),

$$\mathbf{P}\left(\mathbf{X}_i = x_i; \left\{\pi_j\right\}, \theta\right) = \frac{N_i!}{x_{i1}!, \ldots, x_{iK}!} \frac{\prod_{j=1}^{K} \prod_{r=1}^{x_{ij}} \left\{\pi_j \left(1-\theta\right) + \left(r-1\right)\theta\right\}}{\prod_{r=1}^{N_i} \left(1-\theta\right) + \left(r-1\right)\theta}$$

where $\mathbf{x}_i = [x_{i1}, \ldots, x_{iK}]$ is the random vector formed by K taxa (features) counts (RAD vector), $N_i = \sum_{j=1}^{K} x_{ij}$ is the total number of reads (sequence depth), $\{\pi_j\}$ are the mean of taxa-proportions (RAD-probability mean), and θ is the overdispersion parameter.

Note: Though the test statistic supports an unequal number of reads across samples, the performance has not yet been fully tested.

Value

A data matrix of taxa counts where the rows are samples and columns are the taxa.

References

Mosimann, J. E. (1962). On the compound multinomial distribution, the multivariate β-distribution, and correlations among proportions. Biometrika 49, 65-82.
Tvedebrink, T. (2010). Overdispersion in allelic counts and theta-correction in forensic genetics. Theor Popul Biol 78, 200-210.

Examples

```
data(saliva)

### Generate a the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrs <- rep(15000, 20)

### Get gamma from the dirichlet-multinomial parameters
shape <- dirmult(saliva)$gamma

dmData <- Dirichlet.multinomial(nrs, shape)
dmData[1:5, 1:5]
```

DM.MoM *Method-of-Moments (MoM) Estimators of the Dirichlet-Multinomial Parameters*

Description

Method-of-Moments (MoM) estimators of the Dirichlet-multinomial parameters: taxa proportions and overdispersion.

Usage

```
DM.MoM(data)
```

Arguments

`data`	A matrix of taxonomic counts(columns) for each sample(rows).

Details

Given a set of taxa-count vectors $\{\mathbf{x}_i, \ldots, \mathbf{x}_P\}$, the methods of moments (MoM) estimator of the set of parameters θ and $\{\pi_j\}_{j=1}^{K}$ is given as follows (Mosimann, 1962; Tvedebrink, 2010):

$$\hat{\pi}_j = \frac{\sum_{i=1}^{P} x_{ij}}{\sum_{i=1}^{P} N_i},$$

and

$$\hat{\theta} = \sum_{j=1}^{K} \frac{S_j - G_j}{\sum_{j=1}^{K} \left(S_j + (N_c - 1)\, G_j\right)},$$

where $N_c = (P-1)^{-1}\left(\sum_{i=1}^{P} N_i - \left(\sum_{i=1}^{P} N_i\right)^{-1}\sum_{i=1}^{P} N_i^2\right)$, and $S_j = (P-1)^{-1}\sum_{i=1}^{P} N_i\left(\hat{\pi}_{ij} - \hat{\pi}_j\right)^2$, and $G_j = \left(\sum_{i=1}^{P}(N_i - 1)\right)^{-1}\sum_{i=1}^{P} N_i\hat{\pi}_{ij}\left(1 - \hat{\pi}_{ij}\right)$ with $\hat{\pi}_{ij} = \frac{x_{ij}}{N_i}$.

Value

A list providing the MoM estimator for overdispersion, the MoM estimator of the RAD-probability mean vector, and the corresponding loglikelihood value for the given data set and estimated parameters.

References

Mosimann, J. E. (1962). On the compound multinomial distribution, the multivariate β-distribution, and correlations among proportions. Biometrika 49, 65-82.
Tvedebrink, T. (2010). Overdispersion in allelic counts and theta-correction in forensic genetics. Theor Popul Biol 78, 200-210.

Examples

```
data(throat)

fit.throat <- DM.MoM(throat)
fit.throat
```

DM.Rpart *Dirichlet-Multinomial RPart*

Description

This function combines recursive partitioning and the Dirichlet-Multinomial distribution to identify homogeneous subgroups of microbiome taxa count data.

Usage

```
DM.Rpart(data, covars, plot = TRUE, minsplit = 1, minbucket = 1, cp = 0, numCV = 10,
numCon = 100, parallel = FALSE, cores = 3, use1SE = FALSE, lowerSE = TRUE)
```

Arguments

`data` A matrix of taxonomic counts(columns) for each sample(rows).

`covars` A matrix of covariates(columns) for each sample(rows).

`plot` When 'TRUE' a tree plot of the results will be generated.

`minsplit` The minimum number of observations to split on, see rpart.control.

`minbucket` The minimum number of observations in any terminal node, see rpart.control.

`cp` The complexity parameter, see rpart.control.

`numCV` The number folds for a k-fold cross validation. A value less than 2 will return the rpart result without any cross validation.

`numCon` The number of cross validations to repeat to achieve a consensus solution.

`parallel` When this is 'TRUE' it allows for parallel calculation of consensus. Requires the package `doParallel`.

`cores` The number of parallel processes to run if parallel is 'TRUE'.

`use1SE` See details.

`lowerSE` See details.

Details

There are 3 ways to run this function. The first is setting numCV to less than 2, which will run rpart once using the DM distribution and the specified minsplit, minbucket and cp. This result will not have any kind of branch pruning and the objects returned 'fullTree' and 'bestTree' will be the same.

The second way is setting numCV to 2 or greater (we recommend 10) and setting numCon to less than 2. This will run rpart several times using a k-fold cross validation to prune the tree to its optimal size. This is the best method to use.

The third way is setting both numCV and numCon to 2 or greater (We recommend at least 100 for numCon). This will repeat the second way numCon times and build a consensus solution. This method is ONLY needed for low sample sizes.

When the argument 'use1SE' is 'FALSE', the returned object 'bestTree' is the pruned tree with the lowest MSE. When it is 'TRUE', 'bestTree' is either the biggest pruned tree (lowerSE = FALSE) or the smallest pruned tree (lowerSE = TRUE), that is within 1 standard error of the lowest MSE.

Value

The 3 main things returned are:

`fullTree` An rpart object without any pruning.

`bestTree` A pruned rpart object based on use1SE and lowerSE's settings.

`cpTable` Information about the fullTree rpart object and how it splits.

The other variables returned include surrogate/competing splits, error rates and a plot of the bestTree if plot is TRUE.

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Create some covariates for our data set
site <- c(rep("Saliva", nrow(saliva)), rep("Throat", nrow(throat)),
rep("Tonsils", nrow(tonsils)))
covars <- data.frame(Group=site)

### Combine our data into a single object
data <- rbind(saliva, throat, tonsils)
```

```
### For a single rpart tree
numCV <- 0
numCon <- 0
rpartRes <- DM.Rpart(data, covars, numCV=numCV, numCon=numCon)

## Not run:
### For a cross validated rpart tree
numCon <- 0
rpartRes <- DM.Rpart(data, covars, numCon=numCon)

### For a cross validated rpart tree with consensus
numCon <- 2 # Note this is set to 2 for speed and should be at least 100
rpartRes <- DM.Rpart(data, covars, numCon=numCon)

## End(Not run)
```

dmrp_covars *Paper Covariate Set*

Description

This data set is used in the paper Microbiome Recursive Partitioning 2019. It contains 128 subjects and 11 cytokines.

Usage

```
data(dmrp_covars)
```

Format

The format is a data frame of 128 rows by 11 columns, with the each row being a separate subject and each column being a different cytokine.

Examples

```
data(dmrp_covars)
```

dmrp_data *Paper Taxa Data Set*

Description

This data set is used in the paper Microbiome Recursive Partitioning 2019. It contains 128 subjects and 29 genus level taxa.

Usage

```
data(dmrp_data)
```

Format

The format is a data frame of 128 rows by 29 columns, with the each row being a separate subject and each column being a different taxa.

Examples

```
data(dmrp_data)
```

Est.PI *Estimate the Pi Vector*

Description

Calculates Dirichlet-Multinomial parameters for every group using Maximum Likelihood and Method of Moments estimates: Taxa proportion estimates (PI vector) with standard errors and Confidence intervals, as well as theta values with standard errors.

Usage

```
Est.PI(group.data, conf = .95)
```

Arguments

group.data	A list of matrices of taxonomic counts(columns) for each sample(rows).
conf	The desired confidence limits. The default is 95%

Value

A list containing the parameters: PI, SE and the upper/lower bounds of the confidence interval for every taxa, and the theta values with standard errors for both MLE and MOM.

Examples

```
## Not run:
data(saliva)
data(throat)
data(tonsils)

### Combine the data sets into a single list
group.data <- list(saliva, throat, tonsils)

### Get PI using MLE and MOM with CI
piEsts <- Est.PI(group.data)
```

```
mle <- piEsts$MLE
mom <- piEsts$MOM

## End(Not run)
```

`formatDataSets` *Format Data*

Description

For a list of datasets, this function finds the union of taxa across all datasets and transforms them such that they all have the same columns of taxa.

Usage

```
formatDataSets(group.data)
```

Arguments

`group.data`	A list where each element is a matrix of taxonomic counts(columns) for each sample(rows). Note that the row names should correspond to sample names

Details

This function will also sort all the columns into the same order for every dataset and remove any columns that have 0's for every sample.

E.g. For two datasets, any taxa present in dataset1 but not dataset2 will be added to dataset2 with a 0 count for all samples and vice versa.

Value

The list given, but modified so every data set has the same ordering and number of columns

Examples

```
data(saliva)
data(throat)

### Set each data set to have 10 different columns
saliva2 <- saliva[,1:10]
throat2 <- throat[,11:20]

### Combine the data sets into a single list
group.data <- list(saliva2, throat2)

formattedData <- formatDataSets(group.data)
formattedData[[1]][1:5, 1:5]
```

Gen.Alg *Find Taxa Separating Two Groups using Genetic Algorithm (GA)*

Description

GA-Mantel is a fully multivariate method that uses a genetic algorithm to search over possible taxa subsets using the Mantel correlation as the scoring measure for assessing the quality of any given taxa subset.

Usage

```
Gen.Alg(data, covars, iters = 50, popSize = 200, earlyStop = 0,
dataDist = "euclidean", covarDist = "gower", verbose = FALSE,
plot = TRUE, minSolLen = NULL, maxSolLen = NULL, custCovDist = NULL,
penalty = 0)
```

Arguments

`data`	A matrix of taxonomic counts(columns) for each sample(rows).
`covars`	A matrix of covariates(columns) for each sample(rows).
`iters`	The number of times to run through the GA.
`popSize`	The number of solutions to test on each iteration.
`earlyStop`	The number of consecutive iterations without finding a better solution before stopping regardless of the number of iterations remaining. A value of '0' will prevent early stopping.
`dataDist`	The distance metric to use for the data. Either "euclidean" or "gower".
`covarDist`	The distance metric to use for the covariates. Either "euclidean" or "gower".
`verbose`	While 'TRUE' the current status of the GA will be printed periodically.
`plot`	A boolean to plot the progress of the scoring statistics by iteration.
`minSolLen`	The minimum number of columns to select.
`maxSolLen`	The maximum number of columns to select.
`custCovDist`	A custom covariate distance matrix to use in place of calculating one from covars.
`penalty`	A number between 0 and 1 used to penalize the solutions based on the number of selected taxa using the following formula: score - penalty * ((number of selected taxa)/(number of taxa)).

Details

Use a GA approach to find taxa that separate subjects based on group membership or set of covariates.

The data and covariates should be normalized BEFORE use with this function because of distance functions.

This function uses modified code from the rbga function in the genalg package. rbga

Because the GA looks at combinations and uses the raw data, taxa with a small difference in their PIs may be selected and large differences may not be.

The distance calculations use the vegdist package. vegdist

Value

A list containing

`scoreSumm` A matrix summarizing the score of the population. This can be used to figure out if the ga has come to a final solution or not. This data is also plotted if plot is 'TRUE'.

`solutions` The final set of solutions, sorted with the highest scoring first.

`scores` The scores for the final set of solutions.

`time` How long in seconds the ga took to run.

`selected` The selected columns by name.

`nonSelected` The columns that were NOT selected by name.

`selectedIndex` The selected taxa by column number.

Examples

```
## Not run:
data(saliva)
data(throat)

### Combine the data into a single data frame
group.data <- list(saliva, throat)
group.data <- formatDataSets(group.data)
data <- do.call("rbind", group.data)

### Normalize the data by subject
dataNorm <- t(apply(data, 1, function(x){x/sum(x)}))

### Set covars to just be group membership
memb <- c(rep(0, nrow(saliva)), rep(1, nrow(throat)))
covars <- matrix(memb, length(memb), 1)

### We use low numbers for speed. The exact numbers to use depend
### on the data being used, but generally the higher iters and popSize
### the longer it will take to run.  earlyStop is then used to stop the
### run early if the results aren't improving.
iters <- 500
popSize <- 200
earlyStop <- 250

gaRes <- Gen.Alg(dataNorm, covars, iters, popSize, earlyStop)

## End(Not run)
```

Gen.Alg.Consensus *Find Taxa Separating Two Groups using Multiple Genetic Algorithm's (GA) Consensus*

Description

GA-Mantel is a fully multivariate method that uses a genetic algorithm to search over possible taxa subsets using the Mantel correlation as the scoring measure for assessing the quality of any given taxa subset.

Usage

```
Gen.Alg.Consensus(data, covars, consensus = .5, numRuns = 10,
parallel = FALSE, cores = 3, ...)
```

Arguments

`data`	A matrix of taxonomic counts(columns) for each sample(rows).
`covars`	A matrix of covariates(columns) for each sample(rows).
`consensus`	The required fraction (0, 1] of solutions containing an edge in order to keep it.
`numRuns`	Number of runs to do. In practice the number of runs needed varies based on data set size and the GA parameters set.
`parallel`	When this is 'TRUE' it allows for parallel calculation of the bootstraps. Requires the package `doParallel`.
`cores`	The number of parallel processes to run if parallel is 'TRUE'.
`...`	Other arguments for the GA function see Gen.Alg

Details

Use a GA consensus approach to find taxa that separate subjects based on group membership or set of covariates if you cannot run the GA long enough to get a final solution.

Value

A list containing

`solutions`	The best solution from each run.
`consSol`	The consensus solution.
`selectedIndex`	The selected taxa by column number.

Examples

```
## Not run:
data(saliva)
data(throat)

### Combine the data into a single data frame
group.data <- list(saliva, throat)
group.data <- formatDataSets(group.data)
data <- do.call("rbind", group.data)

### Normalize the data by subject
dataNorm <- t(apply(data, 1, function(x){x/sum(x)}))

### Set covars to just be group membership
memb <- c(rep(0, nrow(saliva)), rep(1, nrow(throat)))
covars <- matrix(memb, length(memb), 1)

### We use low numbers for speed. The exact numbers to use depend
### on the data being used, but generally the higher iters and popSize
### the longer it will take to run.  earlyStop is then used to stop the
### run early if the results aren't improving.
iters <- 500
popSize <- 200
earlyStop <- 250
numRuns <- 3

gaRes <- Gen.Alg.Consensus(dataNorm, covars, .5, numRuns, FALSE, 3,
iters, popSize, earlyStop)

## End(Not run)
```

Kullback.Leibler *Kullback Leibler*

Description

Calculates Kullback Leibler divergence for all pairs of the datasets.

Usage

```
Kullback.Leibler(group.data, plot = TRUE, main="Kullback Leibler Divergences",
parallel = FALSE, cores = 3)
```

Arguments

group.data	A list where each element is a matrix of taxonomic counts(columns) for each sample(rows).
plot	When 'TRUE' a heatmap of the results will also be generated.
main	A string to be used as the plots title.

`parallel` When this is 'TRUE' it allows for parallel calculation of the KL distances. Requires the package `doParallel`.

`cores` The number of parallel processes to run if parallel is 'TRUE'.

Value

A matrix of Kullback Leibler divergence values and a heatmap if plot is TRUE.

References

Kotz S, Johnson N.L (1981) Encyclopedia Of Statistical Sciences

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Combine the data sets into a single list
group.data <- list(saliva, throat, tonsils)

## Not run:
kl <- Kullback.Leibler(group.data)
kl

## End(Not run)
```

MC.Xdc.statistics *Size and Power for the Several-Sample DM Parameter Test Comparison*

Description

This Monte-Carlo simulation procedure provides the power and size of the several sample Dirichlet-Multinomial parameter test comparison, using the likelihood-ratio-test statistics.

Usage

```
MC.Xdc.statistics(group.Nrs, numMC = 10, alphap, type = "ha",
siglev = 0.05, est = "mom")
```

Arguments

`group.Nrs` A list specifying the number of reads/sequence depth for each sample in a group with one group per list entry.

`numMC` Number of Monte-Carlo experiments. In practice this should be at least 1,000.

`alphap` If "hnull": A matrix where rows are vectors of alpha parameters for the reference group.
If "ha": A matrix consisting of vectors of alpha parameters for each taxa in each group.

`type` If "hnull": Computes the size of the test.
If "ha": Computes the power of the test. (default)

`siglev` Significance level for size of the test / power calculation. The default is 0.05.

`est` The type of parameter estimator to be used with the Likelihood-ratio-test statistics, 'mle' or 'mom'. Default value is 'mom'. (See Note 2 in details)

Details

1. Note 1: Though the test statistic supports an unequal number of reads across samples, the performance has not yet been fully tested.
2. Note 2: 'mle' will take significantly longer time and may not be optimal for small sample sizes; 'mom' will provide a more conservative result in such a case.
3. Note 3: All components of alphap should be non-zero or it may result in errors and/or invalid results.

Value

Size of the test statistics (under "hnull") or power (under "ha") of the test.

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Get a list of dirichlet-multinomial parameters for the data
fit.saliva <- DM.MoM(saliva)
fit.throat <- DM.MoM(throat)
fit.tonsils <- DM.MoM(tonsils)

### Set up the number of Monte-Carlo experiments
### We use 1 for speed, should be at least 1,000
numMC <- 1

### Generate the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrsGrp1 <- rep(12000, 9)
nrsGrp2 <- rep(12000, 11)
nrsGrp3 <- rep(12000, 12)
group.Nrs <- list(nrsGrp1, nrsGrp2, nrsGrp3)

### Computing size of the test statistics (Type I error)
alphap <- fit.saliva$gamma
pval1 <- MC.Xdc.statistics(group.Nrs, numMC, alphap, "hnull")
pval1
```

```
### Computing Power of the test statistics (Type II error)
alphap <- rbind(fit.saliva$gamma, fit.throat$gamma, fit.tonsils$gamma)
pval2 <- MC.Xdc.statistics(group.Nrs, numMC, alphap)
pval2
```

`MC.Xmc.statistics` *Size and Power of Several Sample RAD-Probability Mean Test Comparison*

Description

This Monte-Carlo simulation procedure provides the power and size of the several sample RAD-probability mean test comparison with known reference vector of proportions, using the Generalized Wald-type statistics.

Usage

```
MC.Xmc.statistics(group.Nrs, numMC = 10, pi0, group.pi, group.theta,
type = "ha", siglev = 0.05)
```

Arguments

`group.Nrs` A list specifying the number of reads/sequence depth for each sample in a group with one group per list entry.

`numMC` Number of Monte-Carlo experiments. In practice this should be at least 1,000.

`pi0` The RAD-probability mean vector.

`group.pi` If `"hnull"`: This argument is ignored.
If `"ha"`: A matrix where each row is a vector pi values for each group.

`group.theta` A vector of overdispersion values for each group.

`type` If `"hnull"`: Computes the size of the test.
If `"ha"`: Computes the power of the test. (default)

`siglev` Significance level for size of the test / power calculation. The default is 0.05.

Details

Note: Though the test statistic supports an unequal number of reads across samples, the performance has not yet been fully tested.

Value

Size of the test statistics (under `"hnull"`) or power (under `"ha"`) of the test.

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Get a list of dirichlet-multinomial parameters for the data
fit.saliva <- DM.MoM(saliva)
fit.throat <- DM.MoM(throat)
fit.tonsils <- DM.MoM(tonsils)

### Set up the number of Monte-Carlo experiments
### We use 1 for speed, should be at least 1,000
numMC <- 1

### Generate the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrsGrp1 <- rep(12000, 9)
nrsGrp2 <- rep(12000, 11)
group.Nrs <- list(nrsGrp1, nrsGrp2)

group.theta <- c(0.01, 0.05)
pi0 <- fit.saliva$pi

### Computing size of the test statistics (Type I error)
pval1 <- MC.Xmc.statistics(group.Nrs, numMC, pi0, group.theta=group.theta, type="hnull")
pval1

### Computing Power of the test statistics (Type II error)
group.pi <- rbind(fit.throat$pi, fit.tonsils$pi)
pval2 <- MC.Xmc.statistics(group.Nrs, numMC, pi0, group.pi, group.theta)
pval2
```

MC.Xmcupo.statistics *Size and Power of Several Sample RAD-Probability Mean Test Comparisons: Unknown Vector of Proportion*

Description

This Monte-Carlo simulation procedure provides the power and size of the several sample RAD-probability mean test comparisons without reference vector of proportions, using the Generalized Wald-type statistics.

Usage

```
MC.Xmcupo.statistics(group.Nrs, numMC = 10, pi0, group.pi, group.theta,
type = "ha", siglev = 0.05)
```

Arguments

`group.Nrs` A list specifying the number of reads/sequence depth for each sample in a group with one group per list entry.

`numMC` Number of Monte-Carlo experiments. In practice this should be at least 1,000.

`pi0` The RAD-probability mean vector.

`group.pi` If `"hnull"`: This argument is ignored.
If `"ha"`: A matrix where each row is a vector pi values for each group.

`group.theta` A vector of overdispersion values for each group.

`type` If `"hnull"`: Computes the size of the test.
If `"ha"`: Computes the power of the test. (default)

`siglev` Significance level for size of the test / power calculation. The default is 0.05.

Details

Note: Though the test statistic supports an unequal number of reads across samples, the performance has not yet been fully tested.

Value

Size of the test statistics (under `"hnull"`) or power (under `"ha"`) of the test.

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Get a list of dirichlet-multinomial parameters for the data
fit.saliva <- DM.MoM(saliva)
fit.throat <- DM.MoM(throat)
fit.tonsils <- DM.MoM(tonsils)

### Set up the number of Monte-Carlo experiments
### We use 1 for speed, should be at least 1,000
numMC <- 1

### Generate the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
Nrs1 <- rep(12000, 10)
Nrs2 <- rep(12000, 19)
group.Nrs <- list(Nrs1, Nrs2)

group.theta <- c(fit.throat$theta, fit.tonsils$theta)
pi0 <- fit.saliva$pi

### Computing size of the test statistics (Type I error)
pval1 <- MC.Xmcupo.statistics(group.Nrs, numMC, pi0, group.theta=group.theta, type="hnull")
pval1
```

```
### Computing Power of the test statistics (Type II error)
group.pi <- rbind(fit.throat$pi, fit.tonsils$pi)
pval2 <- MC.Xmcupo.statistics(group.Nrs, numMC, group.pi=group.pi, group.theta=group.theta)
pval2
```

MC.Xoc.statistics *Size and Power of Several Sample-Overdispersion Test Comparisons*

Description

This Monte-Carlo simulation procedure provides the power and size of the several sample-overdispersion test comparison, using the likelihood-ratio-test statistics.

Usage

```
MC.Xoc.statistics(group.Nrs, numMC = 10, group.alphap, type = "ha", siglev = 0.05)
```

Arguments

`group.Nrs`	A list specifying the number of reads/sequence depth for each sample in a group with one group per list entry.
`numMC`	Number of Monte-Carlo experiments. In practice this should be at least 1,000.
`group.alphap`	If `"hnull"`: A vector of alpha parameters for each taxa. If `"ha"`: A list consisting of vectors of alpha parameters for each taxa.
`type`	If `"hnull"`: Computes the size of the test. If `"ha"`: Computes the power of the test. (default)
`siglev`	Significance level for size of the test / power calculation. The default is 0.05.

Details

1. Note 1: Though the test statistic supports an unequal number of reads across samples, the performance has not yet been fully tested.
2. Note 2: All components of `group.alphap` should be non-zero or it may result in errors and/or invalid results.

Value

Size of the test statistics (under `"hnull"`) or power (under `"ha"`) of the test.

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Get a list of dirichlet-multinomial parameters for the data
fit.saliva <- DM.MoM(saliva)
fit.throat <- DM.MoM(throat)
fit.tonsils <- DM.MoM(tonsils)

### Set up the number of Monte-Carlo experiments
### We use 1 for speed, should be at least 1,000
numMC <- 1

### Generate the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrsGrp1 <- rep(12000, 9)
nrsGrp2 <- rep(12000, 11)
nrsGrp3 <- rep(12000, 12)
group.Nrs <- list(nrsGrp1, nrsGrp2, nrsGrp3)

### Computing size of the test statistics (Type I error)
alphap <- fit.tonsils$gamma
pval1 <- MC.Xoc.statistics(group.Nrs, numMC, alphap, "hnull")
pval1

## Not run:
### Computing Power of the test statistics (Type II error)
alphap <- rbind(fit.saliva$gamma, fit.throat$gamma, fit.tonsils$gamma)
pval2 <- MC.Xoc.statistics(group.Nrs, numMC, alphap, "ha")
pval2

## End(Not run)
```

`MC.Xsc.statistics` *Size and Power for the One Sample RAD Probability-Mean Test Comparison*

Description

This Monte-Carlo simulation procedure provides the power and size of the one sample RAD probability-mean test, using the Generalized Wald-type statistic.

Usage

```
MC.Xsc.statistics(Nrs, numMC = 10, fit, pi0 = NULL, type = "ha", siglev = 0.05)
```

Arguments

`Nrs` A vector specifying the number of reads/sequence depth for each sample.

`numMC` Number of Monte-Carlo experiments. In practice this should be at least 1,000.

`fit` A list (in the format of the output of dirmult function) containing the data parameters for evaluating either the size or power of the test.

`pi0` The RAD-probability mean vector. If the type is set to `"hnull"` then pi0 is set by the sample in fit.

`type`	If `"hnull"`: Computes the size of the test. If `"ha"`: Computes the power of the test. (default)
`siglev`	Significance level for size of the test / power calculation. The default is 0.05.

Details

Note: Though the test statistic supports an unequal number of reads across samples, the performance has not yet been fully tested.

Value

Size of the test statistics (under `"hnull"`) or power (under `"ha"`) of the test.

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Get a list of dirichlet-multinomial parameters for the data
fit.saliva <- DM.MoM(saliva)
fit.throat <- DM.MoM(throat)
fit.tonsils <- DM.MoM(tonsils)

### Set up the number of Monte-Carlo experiments
### We use 1 for speed, should be at least 1,000
numMC <- 1

### Generate the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrs <- rep(15000, 25)

### Computing size of the test statistics (Type I error)
pval1 <- MC.Xsc.statistics(nrs, numMC, fit.tonsils, fit.saliva$pi, "hnull")
pval1

### Computing Power of the test statistics (Type II error)
pval2 <- MC.Xsc.statistics(nrs, numMC, fit.throat, fit.tonsils$pi)
pval2
```

`MC.ZT.statistics` *Size and Power of Goodness of Fit Test: Multinomial vs. Dirichlet-Multinomial*

Description

This Monte-Carlo simulation procedure provides the power and size of the Multinomial vs. Dirichlet-Multinomial goodness of fit test, using the C(α)-optimal test statistics of Kim and Margolin (1992) (t statistics) and the C(α)-optimal test statistics of (Paul et al., 1989).

Usage

```
MC.ZT.statistics(Nrs, numMC = 10, fit, type = "ha", siglev = 0.05)
```

Arguments

`Nrs`	A vector specifying the number of reads/sequence depth for each sample.
`numMC`	Number of Monte-Carlo experiments. In practice this should be at least 1,000.
`fit`	A list (in the format of the output of dirmult function) containing the data parameters for evaluating either the size or power of the test.
`type`	If `"hnull"`: Computes the size of the test. If `"ha"`: Computes the power of the test. (default)
`siglev`	Significance level for size of the test / power calculation. The default is 0.05.

Details

Note: Though the test statistic supports an unequal number of reads across samples, the performance has not yet been fully tested.

Value

A vector containing both the size of the test statistics (under `"hnull"`) or power (under `"ha"`) of the test for both the z and t statistics.

Examples

```
data(saliva)

### Get a list of dirichlet-multinomial parameters for the data
fit.saliva <- DM.MoM(saliva)

### Set up the number of Monte-Carlo experiments
### We use 1 for speed, should be at least 1,000
numMC <- 1

### Generate the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrs <- rep(15000, 25)

### Computing size of the test statistics (Type I error)
pval1 <- MC.ZT.statistics(nrs, numMC, fit.saliva, "hnull")
pval1

### Computing Power of the test statistics (Type II error)
pval2 <- MC.ZT.statistics(nrs, numMC, fit.saliva)
pval2
```

Multinomial *Generation of Multinomial Random Samples*

Description

It generates a data matrix with random samples from a multinomial distribution where the rows are the samples and the columns are the taxa.

Usage

```
Multinomial(Nrs, probs)
```

Arguments

Nrs	A vector specifying the number of reads or sequence depth for each sample.
probs	A vector specifying taxa probabilities.

Details

Note: Though the test statistic supports an unequal number of reads across samples, the performance has not yet been fully tested.

Value

A data matrix of taxa counts where the rows are the samples and the columns are the taxa.

Examples

```
### Generate the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrs <- rep(15000, 25)

### Create a probability vector
probs <- c(0.4, 0.3, 0.2, .05, 0.04, .01)

mData <- Multinomial(nrs, probs)
mData[1:5, 1:5]
```

Plot.MDS *Multidimensional Scaling Plot of Microbiome Data*

Description

Plots any number of data sets on an MDS plot.

Usage

```
Plot.MDS(group.data, main = "Group MDS", retCords = FALSE)
```

Arguments

group.data	A list of matrices of taxonomic counts(columns) for each sample(rows).
main	A string to be used as the plots title.
retCords	A boolean to return the mds coordinates or not.

Value

A MDS plot and possibly the x-y coordinates for every point.

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Combine the data sets into a single list
group.data <- list(saliva, throat, tonsils)

Plot.MDS(group.data)
```

Plot.PI *Plot the Pi Vector*

Description

Plots the taxa proportions for every group.

Usage

```
Plot.PI(estPi, errorBars = TRUE, logScale = FALSE,
main = "PI Vector", ylab = "Fractional Abundance")
```

Arguments

`estPi`	The results for either MLE or MOM from the function 'Est.Pi'.
`errorBars`	A boolean to display the error bars or not.
`logScale`	A boolean to log the y scale or not.
`main`	A string to be used as the plots title.
`ylab`	A string to be used as the plots y-axis title.

Value

A plot of the pi vectors for every group.

Examples

```
## Not run:
data(saliva)
data(throat)
data(tonsils)

### Combine the data sets into a single list
group.data <- list(saliva, throat, tonsils)

### Get PI using MLE with CI
mle <- Est.PI(group.data)$MLE

### Plot with Error Bars
Plot.PI(mle)

### Plot without Error Bars
Plot.PI(mle, FALSE)

### Plot with Error Bars and scaling
Plot.PI(mle, TRUE, TRUE)

## End(Not run)
```

`Plot.RM.Barchart` *Plot the Pi Vector for Repeated Measures*

Description

Plots the taxa proportions for every group/time as a barchart.

Usage

```
Plot.RM.Barchart(group.data, groups, times, plotByGrp = TRUE,
col = NULL, conf = .95)
```

Arguments

`group.data` A list of matrices of taxonomic counts(columns) for each sample(rows).

`groups` A vector indicating group membership.

`times` A vector indicating time.

`plotByGrp` When 'TRUE', the plot will be split by group rather than time.

`col` A vector of colors to use to denote taxa.

`conf` The desired confidence limits. The default is 95%

Value

A barchart of the pi vectors for every group/time.

Examples

```
## Not run:
data(saliva)
data(throat)

### Reduce the size of the data
saliva <- Data.filter(saliva, numTaxa=9)
throat <- Data.filter(throat, numTaxa=9)

### Get the gamma value for the data
saliva.gamma <- DM.MoM(saliva)$gamma
throat.gamma <- DM.MoM(throat)$gamma
mid.gamma <- (saliva.gamma + throat.gamma)/2

### Generate a the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrs <- rep(10000, 20)

### Create data sets to be our time series in a list
group.data <- list(
Dirichlet.multinomial(nrs, saliva.gamma),
Dirichlet.multinomial(nrs, saliva.gamma),
Dirichlet.multinomial(nrs, saliva.gamma),
Dirichlet.multinomial(nrs, saliva.gamma),
Dirichlet.multinomial(nrs, mid.gamma),
Dirichlet.multinomial(nrs, throat.gamma)
)
names(group.data) <- c(
"Group 1, Time 1", "Group 2, Time 1",
"Group 1, Time 2", "Group 2, Time 2",
"Group 1, Time 3", "Group 2, Time 3"
)

### Set the group and time information for each list element
groups <- c(1, 2, 1, 2, 1, 2)
times <- c(1, 2, 3, 1, 2, 3)
```

```
### Plot the data by Group
Plot.RM.Barchart(group.data, groups, times)

### Plot the data by Time
Plot.RM.Barchart(group.data, groups, times, FALSE)

## End(Not run)
```

Plot.RM.Dotplot *Plot the Pi Vector for Repeated Measures*

Description

Plots the taxa proportions for every group/time as a dot plot.

Usage

```
Plot.RM.Dotplot(group.data, groups, times, errorBars = TRUE,
col = NULL, conf = .95, alpha = 1)
```

Arguments

group.data	A list of matrices of taxonomic counts(columns) for each sample(rows).
groups	A vector indicating group membership.
times	A vector indicating time.
errorBars	When 'TRUE', error bars will also be displayed.
col	A vector of colors to use to denote taxa.
conf	The desired confidence limits. The default is 95%
alpha	The desired alpha level for the colors.

Value

A plot of the pi vectors for every group/time.

Examples

```
## Not run:
data(saliva)
data(throat)

### Reduce the size of the data
saliva <- Data.filter(saliva, numTaxa=9)
throat <- Data.filter(throat, numTaxa=9)

### Get the gamma value for the data
saliva.gamma <- DM.MoM(saliva)$gamma
throat.gamma <- DM.MoM(throat)$gamma
```

```
mid.gamma <- (saliva.gamma + throat.gamma)/2

### Generate a the number of reads per sample
### The first number is the number of reads and the second is the number of subjects
nrs <- rep(10000, 20)

### Create data sets to be our time series in a list
group.data <- list(
Dirichlet.multinomial(nrs, saliva.gamma),
Dirichlet.multinomial(nrs, saliva.gamma),
Dirichlet.multinomial(nrs, saliva.gamma),
Dirichlet.multinomial(nrs, saliva.gamma),
Dirichlet.multinomial(nrs, mid.gamma),
Dirichlet.multinomial(nrs, throat.gamma)
)
names(group.data) <- c(
"Group 1, Time 1", "Group 2, Time 1",
"Group 1, Time 2", "Group 2, Time 2",
"Group 1, Time 3", "Group 2, Time 3"
)

### Set the group and time information for each list element
groups <- c(1, 2, 1, 2, 1, 2)
times <- c(1, 2, 3, 1, 2, 3)

### Plot the data with error bars
Plot.RM.Dotplot(group.data, groups, times)

### Plot the data without error bars
Plot.RM.Dotplot(group.data, groups, times, FALSE)

## End(Not run)
```

saliva *Saliva Data Set*

Description

The saliva data set formed by the Ranked-abundance distribution vectors of 24 subjects. The RAD vectors contains 21 elements formed by the 20 most abundant taxa at the genus level and additional taxa containing the sum of the remaining less abundant taxa per sample. Note that the incorporation of the additional taxon (taxon 21) in the analysis allows for estimating the RAD proportional-mean of taxa with respect to all the taxa within the sample.

Usage

```
data(saliva)
```

Format

The format is a matrix of 24 rows by 21 columns, with the each row being a separate subject and each column being a different taxa.

Examples

```
data(saliva)
```

Test.Paired *Test Paired Data Sets*

Description

Tests two paired data sets for similarity.

Usage

```
Test.Paired(group.data, numPerms = 1000, parallel = FALSE, cores = 3)
```

Arguments

group.data	A list of 2 matrices of taxonomic counts(columns) for each sample(rows).
numPerms	Number of permutations. In practice this should be at least 1,000.
parallel	When this is 'TRUE' it allows for parallel calculation of the permutations. Requires the package doParallel.
cores	The number of parallel processes to run if parallel is 'TRUE'.

Value

A pvalue.

Examples

```
data(saliva)
data(throat)


### Since saliva and throat come from same subjects, the data is paired
saliva1 <- saliva[-24,] # Make saliva 23 subjects to match throat
group.data <- list(throat, saliva1)

### We use 1 for speed, should be at least 1,000
numPerms <- 1

pval <- Test.Paired(group.data, numPerms)
pval
```

`throat` *Throat Data Set*

Description

The throat data set formed by the Ranked-abundance distribution vectors of 24 subjects. The RAD vectors contains 21 elements formed by the 20 most abundant taxa at the genus level and additional taxa containing the sum of the remaining less abundant taxa per sample. Note that the incorporation of the additional taxon (taxon 21) in the analysis allows for estimating the RAD proportional-mean of taxa with respect to all the taxa within the sample.

Usage

```
data(throat)
```

Format

The format is a matrix of 24 rows by 21 columns, with the each row being a separate subject and each column being a different taxa.

Examples

```
data(throat)
```

`tongue` *Tongue Data Set*

Description

The tongue data set formed by the Ranked-abundance distribution vectors of 24 subjects. The RAD vectors contains 21 elements formed by the 20 most abundant taxa at the genus level and additional taxa containing the sum of the remaining less abundant taxa per sample. Note that the incorporation of the additional taxon (taxon 21) in the analysis allows for estimating the RAD proportional-mean of taxa with respect to all the taxa within the sample.

Usage

```
data(tongue)
```

Format

The format is a matrix of 24 rows by 21 columns, with the each row being a separate subject and each column being a different taxa.

Examples

```
data(tongue)
```

`tonsils` *Palatine Tonsil Data Set*

Description

The palatine tonsil data set formed by the Ranked-abundance distribution vectors of 24 subjects. The RAD vectors contains 21 elements formed by the 20 most abundant taxa at the genus level and additional taxa containing the sum of the remaining less abundant taxa per sample. Note that the incorporation of the additional taxon (taxon 21) in the analysis allows for estimating the RAD proportional-mean of taxa with respect to all the taxa within the sample.

Usage

```
data(tonsils)
```

Format

The format is a matrix of 24 rows by 21 columns, with the each row being a separate subject and each column being a different taxa.

Examples

```
data(tonsils)
```

`Xdc.sevsample` *Likelihood-Ratio-Test Statistics: Several Sample Dirichlet-Multinomial Test Comparison*

Description

This routine provides the value of the Likelihood-Ratio-Test Statistics and the corresponding p-value for evaluating the several sample Dirichlet-Multinomial parameter test comparison.

Usage

```
Xdc.sevsample(group.data, epsilon = 10^(-4), est = "mom")
```

Arguments

`group.data` A list where each element is a matrix of taxonomic counts(columns) for each sample(rows). (See Notes 1 and 2 in details)

`epsilon` Convergence tolerance. To terminate, the difference between two succeeding log-likelihoods must be smaller than epsilon. Default value is 10^(-4).

`est` The type of parameter estimator to be used with the Likelihood-ratio-test statistics, 'mle' or 'mom'. Default value is 'mom'. (See Note 3 in details)

Details

To assess whether the Dirichlet parameter vector, $\alpha_{\mathrm{m}} = \pi_{\mathrm{m}} \frac{1-\theta_{\mathrm{m}}}{\theta_{\mathrm{m}}}$ (a function of the RAD probability-mean vector and overdispersion), observed in J groups of microbiome samples are equal to each other, the following hypothesis $\mathrm{H_o} : \alpha_1 = \cdots = \alpha_{\mathrm{m}} = \cdots = \alpha_{\mathrm{J}} = \alpha_{\mathrm{o}}$ versus $\mathrm{H_a} : \alpha_{\mathrm{m}} \neq \alpha_{\mathrm{o}}, m = 1, \ldots, J$ can be tested. The null hypothesis implies that the HMP samples across groups have the same mean and overdispersion, indicating that the RAD models are identical. In particular, the likelihood-ratio test statistic is used, which is given by,

$$x_{\mathrm{dc}} = -2 \log \left\{ \frac{L\left(\alpha_{\mathrm{o}}; \mathbf{X}_1, \ldots, \mathbf{X}_{\mathrm{J}}\right)}{L\left(\alpha_1, \ldots, \alpha_{\mathrm{J}}; \mathbf{X}_1, \ldots, \mathbf{X}_{\mathrm{J}}\right)} \right\}.$$

The asymptotic null distribution of x_{dc} follows a Chi-square with degrees of freedom equal to (J-1)*K, where K is the number of taxa (Wilks, 1938).

1. Note 1: The matrices in `group.data` must contain the same taxa, in the same order.
2. Note 2: Each taxa should be present in at least 1 sample, a column with all 0's may result in errors and/or invalid results.
3. Note 3: 'mle' will take significantly longer time and may not be optimal for small sample sizes; 'mom' will provide more conservative results in such a case.

Value

A list containing the Xdc statistics and p-value.

References

Wilks, S. S. (1938). The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. The Annals of Mathematical Statistics 9, 60-62.

Examples

```
data(saliva)
data(throat)

### Combine the data sets into a single list
group.data <- list(saliva, throat)

xdc <- Xdc.sevsample(group.data)
xdc
```

`Xmc.sevsample` *Generalized Wald-type Statistics: Several Sample RAD Probability-Mean Test Comparison with a Known Common Vector*

Description

This function computes the Generalized Wald-type test statistic (Wilson and Koehler, 1984) and corresponding p-value to assess whether the sample RAD probability-means from multiple populations are the same or different. The statistics assumes that a common RAD probability-mean vector for comparison under the null hypothesis is known.

Usage

```
Xmc.sevsample(group.data, pi0)
```

Arguments

group.data	A list where each element is a matrix of taxonomic counts(columns) for each sample(rows).
pi0	The RAD-probability mean vector.

Details

Note: The matrices in `group.data` must contain the same taxa, in the same order.

Value

A list containing the Generalized Wald-type statistics and p-value.

References

Wilson, J. R., and Koehler, K. J. (1984). Testing of equality of vectors of proportions for several cluster samples. Proceedings of Joint Statistical Association Meetings. Survey Research Methods.

Examples

```
data(saliva)
data(throat)
data(tonsils)

### Get pi from the dirichlet-multinomial parameters
pi0 <- dirmult(saliva)$pi

### Combine the data sets into a single list
group.data <- list(throat, tonsils)

xmc <- Xmc.sevsample(group.data, pi0)
xmc
```

Xmcupo.effectsize *Effect Size for Xmcupo Statistic*

Description

This function computes the Cramer's Phi and Modified Cramer's Phi Criterion for the test statistic `Xmcupo.sevsample`.

Usage

```
Xmcupo.effectsize(group.data)
```

Arguments

group.data A list where each element is a matrix of taxonomic counts(columns) for each sample(rows).

Details

Note: The matrices in group.data must contain the same taxa, in the same order.

Value

A vector containing the Chi-Squared statistic value, the Cramer's Phi Criterion, and the modified Cramer's Phi Criterion.

Examples

```
data(saliva)
data(throat)

### Combine the data sets into a single list
group.data <- list(saliva, throat)

effect <- Xmcupo.effectsize(group.data)
effect
```

Xmcupo.sevsample *Generalized Wald-type Statistics: Several Sample RAD Probability-Mean Test Comparison with an Unknown Common Vector*

Description

This function computes the Generalized Wald-type test statistic (Wilson and Koehler, 1984) and corresponding p-value to assess whether the sample RAD probability-means from multiple populations are same or different. The statistics assumes that a common RAD probability-mean vector for comparison under the null hypothesis is unknown.

Usage

```
Xmcupo.sevsample(group.data)
```

Arguments

group.data A list where each element is a matrix of taxonomic counts(columns) for each sample(rows).

Details

Note: The matrices in group.data must contain the same taxa, in the same order.

Value

A list containing the Generalized Wald-type statistics and p-value.

References

Wilson, J. R., and Koehler, K. J. (1984). Testing of equality of vectors of proportions for several cluster samples. Proceedings of Joint Statistical Association Meetings. Survey Research Methods.

Examples

```
data(saliva)
data(tonsils)
data(throat)

### Combine the data sets into a single list
group.data <- list(saliva, throat, tonsils)

xmcupo <- Xmcupo.sevsample(group.data)
xmcupo
```

`Xoc.sevsample` *Likelihood-Ratio-Test Statistics: Several Sample Overdispersion Test Comparison*

Description

This routine provides the value of the likelihood-ratio-test statistic and the corresponding p-value to assess whether the overdispersion observed in multiple groups of microbiome samples are equal.

Usage

```
Xoc.sevsample(group.data, epsilon = 10^(-4))
```

Arguments

`group.data`	A list where each element is a matrix of taxonomic counts(columns) for each sample(rows). (See Notes 1 and 2 in details)
`epsilon`	Convergence tolerance. To terminate, the difference between two succeeding log-likelihoods must be smaller than epsilon. Default value is 10^(-4).

Details

To assess whether the over dispersion parameter vectors θ_{m} observed in J groups of microbiome samples are equal to each other, the following hypothesis $\mathrm{H_o} : \theta_1 = \cdots = \theta_{\mathrm{m}} = \cdots = \theta_{\mathrm{J}} = \theta_{\mathrm{o}}$ versus $\mathrm{H_a} : \theta_{\mathrm{m}} \neq \theta_{\mathrm{o}}, m = 1, \ldots, J$ can be tested. In particular, the likelihood-ratio test statistic is used (Tvedebrink, 2010), which is given by,

$$x_{\mathrm{oc}} = -2 \log \left\{ \frac{L\left(\theta_{\mathrm{o}}; \mathbf{X}_1, \ldots, \mathbf{X}_{\mathrm{J}}\right)}{L\left(\theta_1, \ldots, \theta_{\mathrm{J}}; \mathbf{X}_1, \ldots, \mathbf{X}_{\mathrm{J}}\right)} \right\}.$$

The asymptotic null distribution of x_{oc} follows a Chi-square with degrees of freedom equal to (J-1) (Wilks, 1938).

1. Note 1: The matrices in `group.data` must contain the same taxa, in the same order.
2. Note 2: Each taxa should be present in at least 1 sample, a column with all 0's may result in errors and/or invalid results.

Value

A list containing the Xoc statistics and p-value.

References

Tvedebrink, T. (2010). Overdispersion in allelic counts and theta-correction in forensic genetics. Theor Popul Biol 78, 200-210.
Wilks, S. S. (1938). The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. The Annals of Mathematical Statistics 9, 60-62.

Examples

```
data(saliva)
data(tonsils)

### Combine the data sets into a single list
group.data <- list(saliva, tonsils)

## Not run:
xoc <- Xoc.sevsample(group.data)
xoc

## End(Not run)
```

Xsc.onesample *Generalized Wald-Type Statistics: One Sample RAD Probability-Mean Test Comparison*

Description

This routine provides the value of the Generalized Wald-type statistic to assess whether the RAD probability-mean observed in one group of samples is equal to a known RAD probability-mean.

Usage

```
Xsc.onesample(data, pi0)
```

Arguments

`data`	A matrix of taxonomic counts(columns) for each sample(rows).
`pi0`	The RAD-probability mean vector.

Value

A list containing Generalized Wald-type statistics and p-value.

Examples

```
data(saliva)
data(throat)

### Get pi from the dirichlet-multinomial parameters
pi0 <- dirmult(saliva)$pi

xsc <- Xsc.onesample(throat, pi0)
xsc
```

Index