Package 'PERFect'

January 18, 2021

Type Package

Title Permutation filtration for microbiome data

Version 1.4.0

Date 2019-09-02

Author Ekaterina Smirnova <ekaterina.smirnova@vcuhealth.org>,
Quy Cao <quy.cao@umontana.edu>

Maintainer Quy Cao <quy.cao@umontana.edu>

Description PERFect is a novel permutation filtering approach designed to address two unsolved problems in microbiome data processing: (i) define and quantify loss due to filtering by implementing thresholds, and (ii) introduce and evaluate a permutation test for filtering loss to provide a measure of excessive filtering.

Depends R (>= 3.6.0), sn (>= 1.5.2)

Imports ggplot2 (>= 3.0.0), phyloseq (>= 1.28.0), zoo (>= 1.8.3), psych (>= 1.8.4), stats (>= 3.6.0), Matrix (>= 1.2.14), fitdistrplus (>= 1.0.12), parallel (>= 3.6.0)

License Artistic-2.0

Encoding UTF-8

LazyData false

BugReports https://github.com/cxquy91/PERFect/issues

URL https://github.com/cxquy91/PERFect

Suggests knitr, rmarkdown, kableExtra, ggpubr

biocViews Software, Microbiome, Sequencing, Classification, Metagenomics

VignetteBuilder knitr

RoxygenNote 7.1.0

NeedsCompilation no

git_url https://git.bioconductor.org/packages/PERFect

git_branch RELEASE_3_12

git_last_commit 69a0a0a

git_last_commit_date 2020-10-27

Date/Publication 2021-01-17

R topics documented:

`DiffFiltLoss` *Difference in filtering loss*

Description

This function calculates differences in filtering loss due to removing a set of J taxa sequentially.

Usage

```
DiffFiltLoss(X, Order_Ind, Plot = TRUE, Taxa_Names = NULL)
```

Arguments

`X`	OTU table, where taxa are columns and samples are rows of the table. It should be a in dataframe format with columns corresponding to taxa names.
`Order_Ind`	Numeric column order corresponding to taxa importance arrangement.
`Plot`	A binary TRUE/FALSE value. If TRUE, the function returns plot of sequential differences in filtering loss.
`Taxa_Names`	Optional taxa labels corresponding to the columns ordering given by Order_Ind.

Details

This function calculates and plots (if Plot = TRUE) differences in filtering loss sequentially for removing the first j taxa as DFL(j+1) = FL(J_j+1) - FL(J_j) for taxa j=1, ..., p.

Value

`DFL`	Differences in filtering loss values.
`p_FL`	Plot of the differences in filtering loss.

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. "PERFect: permutation filtration of microbiome data', to be submitted.

See Also

FiltLoss

Examples

```
data(mock2)

# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

#arrange counts in order of increasing number of samples taxa are present in
NP <- NP_Order(Counts)

#obtain numeric column order corresponding to taxa importance arrangment
Order_Ind <- match(NP, names(Prop))
DFL <- DiffFiltLoss(X=Prop, Order_Ind = Order_Ind, Plot = TRUE, Taxa_Names = NP)

#Differences in filtering loss values
DFL$DFL

#Plot of the differences in filtering loss
DFL$p_FL
```

FiltLoss *Filtering Loss*

Description

Sequential filtering loss calculation for removing a set of J_j taxa for J= 1, ..., p.

Usage

```
FiltLoss(X, Order = "NP", Order.user = NULL, type = "Cumu", Plot = TRUE)
```

Arguments

X	OTU table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.
Order	Taxa ordering. The default ordering is the number of occurrences (NP) of the taxa in all samples. Other types of order are number of connected taxa and weighted number of connected taxa, denoted as "NC", "NCw" respectively. More details about taxa ordering are described in Smirnova et al. User can also specify their preference order with Order.user.

`Order.user`	User's taxa ordering. This argument takes a character vector of ordered taxa names.
`type`	Type of filtering loss calculation.
	`"Ind"` Individual taxon's filtering loss FL_u(j)
	`"Cumu"` Cumulative filtering loss FL(J) due to removing a set of taxa J
`Plot`	Binary TRUE/FALSE value. If TRUE, the function returns plot of sequential differences in filtering loss.

Details

The individual filtering loss due to removing one taxon j is defined as:

FL_u(j)= 1- (||X^T_-j X_-j||_F^2/||X^TX||_F^2),

where X_-j is the matrix X without column corresponding to jth taxon and ||Z||_F is the Frobenious norm of a matrix Z.

The cumulative filtering loss due to removing a set of taxa is defined as:

FL(J)= 1- (||X^T_-J X_-J||_F^2\||X^TX||_F^2),

where X_-J is the n x (p-|J|) dimensional matrix obtained by removing the columns indexed by the set J from the data matrix X.

The cumulative filtering loss is calculated sequentially for each set of taxa J_j, j=1, ..., p.

Value

`FL`	Filtering loss values.
`p_FL`	Plot of filtering loss values.

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. "PERFect: permutation filtration of microbiome data", to be submitted.

See Also

`DiffFiltLoss`

Examples

```
data(mock2)

# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

#Calculate cumulative filtering loss
FL <- FiltLoss(X=Prop, Order = "NP", type = "Cumu", Plot = TRUE)
```

```
#Differences in filtering loss values
FL$FL

#Plot of the differences in filtering loss
FL$p_FL
```

FL_J *Filtering Loss for a set of filtered taxa J*

Description

This function calculates filtering loss due to removing a group of J taxa.

Usage

```
FL_J(X, J)
```

Arguments

X	OTU table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.
J	A vector of J taxa to be removed. It must be subset of column names of X.

Value

FL Filtering loss value.

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutation filtration of microbiome data", to be submitted.

See Also

FiltLoss

Examples

```
data(mock2)

# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

#arrange counts in order of increasing number of samples taxa are present in
NP <- NP_Order(Counts)
Counts <- Counts[,NP]
```

```
# Extract the taxa names to be removed
J <- colnames(Counts)[1:30]

#Calculate filtering loss due to removing these taxa
FL_J(Counts,J)
```

---

mock2 *Bias experiment data*

Description

These publicly available data (Brooks et al., 2015) were generated as a part of a study designed to evaluate the bias at each step of the VCU sequencing protocol, namely, DNA extraction, PCR amplification, sequencing and taxonomic classification. Mock community samples were created out of 7 vaginally relevant bacteria by mixing prescribed quantities of cells, with quantities varying across samples according to an experimental design described in Brooks et al, 2015. As opposed to the positive controls data, bacteria appear in different proportions across samples. The number of taxa identified by the sequencing and bioinformatics pipeline was 46.

Usage

```
data(mock2)
```

Format

This file contains a count OTU table and a proportion OTU table, each with 240 samples and 46 taxa. A list of true taxa is also given.

NCw_Order *Taxa importance ordering by the weighted number of connected taxa*

Description

Taxa importance ordering by the weighted number of connected taxa

Usage

```
NCw_Order(Counts)
```

Arguments

Counts	OTU COUNTS table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.

Value

NCW Taxa names in increasing order of the weighted number of connected taxa.

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutation filtration of microbiome data", to be submitted.

Examples

```
data(mock2)
# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

#arrange counts in order of increasing number of samples taxa are present in
NCw <- NCw_Order(Counts)
```

NC_Order *Taxa importance ordering by the number of connected taxa*

Description

Taxa importance ordering by the number of connected taxa

Usage

```
NC_Order(Counts)
```

Arguments

Counts	OTU COUNTS table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.

Value

NC Taxa names in increasing order of the number of connected taxa.

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutation filtration of microbiome data", to be submitted.

Examples

```
data(mock2)
# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

#arrange counts in order of increasing number of samples taxa are present in
NC <- NC_Order(Counts)
```

NP_Order *Taxa importance ordering by the number of occurrences of the taxa in the n samples*

Description

Taxa importance ordering by the number of occurrences of the taxa in the n samples

Usage

```
NP_Order(Counts)
```

Arguments

Counts	OTU COUNTS table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.

Value

NP Taxa names in increasing order of the number of samples taxa are present in.

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutation filtration of microbiome data", to be submitted.

Examples

```
data(mock2)
# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

#arrange counts in order of increasing number of samples taxa are present in
NP <- NP_Order(Counts)
```

PERFect_perm *Permutation PERFect filtering for microbiome data*

Description

Permutation filtering of the provided OTU table X at a test level alpha. Each set of j taxa significance is evaluated by fitting the Skew-Normal, Normal, t or Cauchy distribution to the sampling distribution obtained by permuted taxa labels.

Usage

```
PERFect_perm(X, infocol = NULL, Order = "NP", Order.user = NULL, normalize = "counts",
    algorithm = "fast", center = FALSE, quant = c(0.1, 0.25, 0.5),
   distr = "sn", alpha = 0.1, rollmean = TRUE, direction = "left", pvals_sim = NULL,
    k = 10000, nbins = 30, hist = TRUE, col = "red", fill = "green",
    hist_fill = 0.2, linecol = "blue")
```

Arguments

X	OTU table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.
infocol	Index vector of the metadata. We assume user only gives a taxa table, but if the metadata of the samples are included in the columns of the input, this option needs to be specified.
Order	Taxa ordering. The default ordering is the number of occurrences (NP) of the taxa in all samples. Other types of order are p-value ordering, number of connected taxa and weighted number of connected taxa, denoted as "pvals", "NC", "NCw" respectively. More details about taxa ordering are described in Smirnova et al. User can also specify their preference order with Order.user.
Order.user	User's taxa ordering. This argument takes a character vector of ordered taxa names.
normalize	Normalizing taxa count. The default option does not normalize taxa count, but user can convert the OTU table into a proportion table using the option "prop" or convert it into a presence/absence table using "pres".
algorithm	Algorithm speed. The default is speed is "fast", which allows the program to efficiently search for significant taxa without computing all the p-values. User must use the default option "hist = FALSE" for the fast algorithm. The alternative setting is "full", which computes all the taxa's p-values.
center	Centering OTU table. The default option does not center the OTU table.
quant	Quantile values used to fit the distribution to log DFL values. The number of quantile values corresponds to the number of parameters in the distribution the data is fitted to. Assuming that at least 50% of taxa are not informative, we suggest fitting the log Skew-Normal distribution by matching the 10%, 25% and 50% percentiles of the log-transformed samples to the Skew-Normal distribution.
distr	The type of distribution to fit log DFL values to. While we suggest using Skew-Normal distribution, and set as the default distribution, other choices are available.

	"sn" Skew-Normal distribution with 3 parameters: location xi, scale omega^2 and shape alpha
	"norm" Normal distribution with 2 parameters: mean and standard deviation sd
alpha	Test level alpha, set to 0.1 by default.
rollmean	Binary TRUE/FALSE value. If TRUE, rolling average (moving mean) of p-values will be calculated, with the lag window set to 3 by default.
direction	Character specifying whether the index of the result should be left- or right-aligned or centered compared to the rolling window of observations, set to "left" by default.
pvals_sim	Object resulting from simultaneous PERFect with taxa abundance ordering, allowing user to perform Simultaneous PERFect with p-values ordering. Be aware that the choice of distribution for both methods must be the same.
k	The number of permutations, set to 10000 by default.
nbins	Number of bins used to visualize the histogram of log DFL values, set to 30 by default.
hist	Binary TRUE/FALSE value. If TRUE, the function builds histograms for each taxon.
col	Graphical parameter for color of histogram bars border, set to "red" by default.
fill	Graphical parameter for color of histogram fill, set to "green" by default.
hist_fill	Graphical parameter for intensity of histogram fill, set to 0.2 by default.
linecol	Graphical parameter for the color of the fitted distribution density, set to "blue" by default.

Details

Filtering is the process of identifying and removing a subset of taxa according to a particular criterion. As opposed to the the simultaneous filtering approach, we do not assume that all distributions for each set of taxa are identical and equal to the distribution of simultaneous filtering. Function PERFect_perm() filters the provided OTU table X and outputs a filtered table that contains signal taxa. PERFect_perm() calculates differences in filtering loss DFL for each taxon according to the given taxa order. By default, the function fits Skew-Normal distribution to the log-differences in filtering loss but Normal, t, or Cauchy distributions can be also used.

Value

If "algorithm = full" is chosen, a list is returned containing:

filtX	Filtered OTU table.
info	The metadata information.
pvals	P-values of the test.
DFL	Differences in filtering loss values.
fit	Fitted values and further goodness of fit details passed from the fitdistr() function.
hist	Histogram of log differences in filtering loss.
est	Estimated distribution parameters.
dfl_distr	Plot of differences in filtering loss values.

If "algorithm = fast" is chosen, fit, hist, est, dfl_distr will not be returned.

Author(s)

Ekaterina Smirnova

References

Azzalini, A. (2005). The skew-normal distribution and related multivariate families. Scandinavian Journal of Statistics, 32(2), 159-188.

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutationfiltration of microbiome data", to be submitted.

See Also

PERFect_sim

Examples

```
data(mock2)

# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

# Perform simultaenous filtering of the data
res_sim <- PERFect_sim(X=Counts)

#order according to p-values
pvals_sim <- pvals_Order(Counts, res_sim)

## Not run:
# obtain permutation PERFEct results using NP taxa ordering
res_perm <- PERFect_perm(X = Prop, Order.user = pvals_sim, algorithm = "fast")

# permutation perfect colored by FLu values
pvals_Plots(PERFect = res_perm, X = Counts, quantiles = c(0.25, 0.5, 0.8, 0.9), alpha=0.05)

## End(Not run)
```

PERFect_perm_reorder *Permutation PERFect filtering for microbiome data*

Description

This function filters the provided OTU table X at a test level alpha given a fitted object perfect_perm obtained by running PERFect_perm() function. PERFect_perm_reorder() reavaluates taxa significance p-values for a different taxa ordering.

Usage

```
PERFect_perm_reorder(X, Order = "NP", Order.user = NULL, res_perm, normalize = "counts",
   center = FALSE, alpha = 0.1, distr = "sn", rollmean = TRUE, direction = "left",
    pvals_sim = NULL)
```

Arguments

X	OTU table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.
Order	Taxa ordering. The default ordering is the number of occurrences (NP) of the taxa in all samples. Other types of order are p-value ordering, number of connected taxa and weighted number of connected taxa, denoted as "pvals", "NC", "NCw" respectively. More details about taxa ordering are described in Smirnova et al. User can also specify their preference order with Order.user.
Order.user	User's taxa ordering. This argument takes a character vector of ordered taxa names.
res_perm	Output of PERFect_perm() function.
normalize	Normalizing taxa count. The default option does not normalize taxa count, but user can convert the OTU table into a proportion table using the option "prop" or convert it into a presence/absence table using "pres".
center	Centering OTU table. The default option does not center the OTU table.
alpha	Test level alpha, set to 0.1 by default.
distr	The type of distribution used in PERFect_perm() function to obtain res_perm object.
	"sn" Skew-Normal distribution with 3 parameters: location xi, scale omega^2 and shape alpha
	"norm" Normal distribution with 2 parameters: mean and standard deviation sd
	"t" Student t-distribution with 2 parameters: n degrees of freedom and noncentrality ncp
	"cauchy" Cauchy distribution with 2 parameters: location and scale
rollmean	Binary TRUE/FALSE value. If TRUE, rolling average (moving mean) of p-values will be calculated, with the lag window set to 3 by default.
direction	Character specifying whether the index of the result should be left- or right-aligned or centered compared to the rolling window of observations, set to "left" by default.
pvals_sim	Object resulting from simultaneous PERFect with taxa abundance ordering, allowing user to perform Simultaneous PERFect with p-values ordering. Be aware that the choice of distribution for both methods must be the same.

Details

This function is designed to save computational time needed to obtain and fit the sampling distribution for each taxon if taxa ordering different from the one used in PERFect_perm() is used. Note, the distribution and OTU table X should match the distribution used in PERFect_perm().

Value

res_perm	The perfect_perm object updated according to the alternative taxa ordering. All elements in this list are same as in perfect_perm object given by PERFect() function.

Author(s)

Ekaterina Smirnova

References

Azzalini, A. (2005). The skew-normal distribution and related multivariate families. Scandinavian Journal of Statistics, 32(2), 159-188.

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutationfiltration of microbiome data", to be submitted.

See Also

PERFect_perm

Examples

```
data(mock2)

# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

## Not run:
# obtain permutation PERFEct results using NP taxa ordering
system.time(res_perm <- PERFect_perm(X=Prop, k = 1000, algorithm = "fast"))

# run PERFEct_sim() function and obtain p-values ordering
res_sim <- PERFect_sim(X=Prop)

# order according to p-values
pvals_sim <- pvals_Order(Counts, res_sim)

# update perfect_perm object according to p-values ordering
res_reorder <- PERFect_perm_reorder(X=Prop,  Order.user = pvals_sim,  res_perm = res_perm)

# permutation perfect colored by FLu values
pvals_Plots(PERFect = res_perm, X = Counts, quantiles = c(0.25, 0.5, 0.8, 0.9), alpha=0.05)

## End(Not run)
```

PERFect_sim *Simulation PERFect filtering for microbiome data*

Description

Simultaneous filtering of the provided OTU table X at a test level alpha. One distribution is fit to taxa simultaneously.

Usage

```
PERFect_sim(X,infocol = NULL, Order = "NP", Order.user = NULL, normalize = "counts",
          center = FALSE, quant = c(0.1, 0.25, 0.5), distr = "sn",
          alpha = 0.1, rollmean = TRUE, direction = "left", pvals_sim = NULL,
          nbins = 30, col = "red", fill = "green", hist_fill = 0.2,
          linecol = "blue")
```

Arguments

X	OTU table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names. It could contains columns of metadata.
infocol	Index vector of the metadata. We assume user only gives a taxa table, but if the metadata of the samples are included in the columns of the input, this option needs to be specified.
Order	Taxa ordering. The default ordering is the number of occurrences (NP) of the taxa in all samples. Other types of order are p-value ordering, number of connected taxa and weighted number of connected taxa, denoted as "pvals", "NC", "NCw" respectively. More details about taxa ordering are described in Smirnova et al. User can also specify their preference order with Order.user.
Order.user	User's taxa ordering. This argument takes a character vector of ordered taxa names.
normalize	Normalizing taxa count. The default option does not normalize taxa count, but user can convert the OTU table into a proportion table using the option "prop" or convert it into a presence/absence table using "pres".
center	Centering OTU table. The default option does not center the OTU table.
quant	Quantile values used to fit the distribution to log DFL values. The number of quantile values corresponds to the number of parameters in the distribution the data is fitted to. Assuming that at least 50% of taxa are not informative, we suggest fitting the log Skew-Normal distribution by matching the 10%, 25% and 50% percentiles of the log-transformed samples to the Skew-Normal distribution.
distr	The type of distribution to fit log DFL values to. While we suggest using Skew-Normal distribution, and set as the default distribution, other choices are available. "sn" Skew-Normal distribution with 3 parameters: location xi, scale omega^2 and shape alpha "norm" Normal distribution with 2 parameters: mean and standard deviation sd "t" Student t-distribution with 2 parameters: n degrees of freedom and noncentrality ncp "cauchy" Cauchy distribution with 2 parameters: location and scale
alpha	Test level alpha, set to 0.1 by default.
rollmean	Binary TRUE/FALSE value. If TRUE, rolling average (moving mean) of p-values will be calculated, with the lag window set to 3 by default.
direction	Character specifying whether the index of the result should be left- or right-aligned or centered compared to the rolling window of observations, set to "left" by default.
pvals_sim	Object resulting from simultaneous PERFect with taxa abundance ordering, allowing user to perform Simultaneous PERFect with p-values ordering. Be aware that the choice of distribution for both methods must be the same.
nbins	Number of bins used to visualize the histogram of log DFL values, set to 30 by default.
col	Graphical parameter for color of histogram bars border, set to "red" by default.
fill	Graphical parameter for color of histogram fill, set to "green" by default.
hist_fill	Graphical parameter for intensity of histogram fill, set to 0.2 by default.
linecol	Graphical parameter for the color of the fitted distribution density, set to "blue" by default.

Details

Filtering is the process of identifying and removing a subset of taxa according to a particular criterion. Function `PERFect_sim()` filters the provided OTU table X and outputs a filtered table that contains signal taxa. `PERFect_sim()` calculates differences in filtering loss DFL for each taxon according to the given taxa order. By default, the function fits Skew-Normal distribution to the log-differences in filtering loss but Normal, t, or Cauchy distributions can be also used. This is implementation of Algorithm 1 described in Smirnova et al.

Value

A list is returned containing:

`filtX`	Filtered OTU table.
`info`	The metadata information.
`pvals`	P-values of the test.
`DFL`	Differences in filtering loss values.
`fit`	Fitted values and further goodness of fit details passed from the `fitdistr()` function.
`hist`	Histogram of log differences in filtering loss.
`est`	Estimated distribution parameters.
`pDFL`	Plot of differences in filtering loss values.

Author(s)

Ekaterina Smirnova

References

Azzalini, A. (2005). The skew-normal distribution and related multivariate families. Scandinavian Journal of Statistics, 32(2), 159-188.

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutationfiltration of microbiome data", to be submitted.

See Also

`PERFect_perm`

Examples

```
data(mock2)
# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts
dim(Counts) # 240x46

# Perform simultaenous filtering of the data
res_sim <- PERFect_sim(X=Counts)
dim(res_sim$filtX)      # 240x10, removing 36 taxa
colnames(res_sim$filtX) # signal taxa
```

```
#permutation perfect colored by FLu values
pvals_Plots(PERFect = res_sim, X = Counts, quantiles = c(0.25, 0.5, 0.8, 0.9), alpha=0.05)
```

`pvals_Order` *Taxa importance ordering by PERFect p-values*

Description

This function orders taxa by increasing significance of simultaneous PERFect p-values.

Usage

```
pvals_Order(Counts, res_sim)
```

Arguments

`Counts`	OTU COUNTS table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.
`res_sim`	Output of `PERFect_sim()` function.

Value

Order_pvals Taxa names in increasing order of p-values significance.

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutation filtration of microbiome data", to be submitted.

See Also

`PERFect_sim`, `PERFect_perm`

Examples

```
data(mock2)

# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts

# Perform simultaenous filtering of the data
res_sim <- PERFect_sim(X=Counts)

#order according to p-values
pvals_sim <- pvals_Order(Counts, res_sim)
```

pvals_Plots *Plots of PERFect p-values*

Description

Graphical representation of p-values obtained by running PERFect_sim() or PERFect_perm() for jth taxon colored by quantile values of individual filtering loss.

Usage

```
pvals_Plots(PERFect, X, quantiles = c(0.25, 0.5, 0.8, 0.9), alpha = 0.1)
```

Arguments

PERFect	Output of PERFect_sim() or PERFect_perm() function.
X	OTU table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.
quantiles	Quantile values for coloring, these are set to 25%, 50%, 80% and 90% percentiles of the individual filtering loss values.
alpha	Alpha level of the test, set to 0.1 by default.

Value

A list is returned containing:

df	Dataframe of taxa names, p-values, Flu values and quantiles.
p_vals	Plot of p-values.

Author(s)

Ekaterina Smirnova

See Also

PERFect_sim, PERFect_perm

Examples

```
data(mock2)
# Proportion data matrix
Prop <- mock2$Prop

# Counts data matrix
Counts <- mock2$Counts
dim(Counts) # 240x46

# Perform simultaenous filtering of the data
res_sim <- PERFect_sim(X=Counts)
dim(res_sim$filtX)      # 240x10, removing 36 taxa
colnames(res_sim$filtX) # signal taxa

#permutation perfect colored by FLu values
pvals_Plots(PERFect = res_sim, X = Counts, quantiles = c(0.25, 0.5, 0.8, 0.9), alpha=0.05)
```

TraditR1 *Traditional Filtering Rule 1*

Description

This rule suggests to remove taxa that are mostly absent in all samples.

Usage

```
TraditR1(X, thresh =5)
```

Arguments

X	OTU COUNTS table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.
thresh	Numerical value, set to 5 by default. Throughout all samples, taxa that are present for less than this threshold with be removed.

Value

filtX	Filtered OTU table

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. "PERFect: permutation filtration of microbiome data.

Examples

```
data(mock2)

# Counts data matrix
Counts <- mock2$Counts

# Filtering
Filtered_X <- TraditR1(Counts)
```

TraditR2 *Traditional Filtering Rule 2*

Description

This rule is adopted from Milici et al. (2016) that removes taxa with low abundance level. Specifically, it keeps taxa with abundance level higher than 0.001%. Then it further selects taxa that satisfy at least one of the following conditions: Present in at least one sample at a relative abundance higher than 1% of the reads of that sample, present in at least 2% of samples at a relative abundance higher than 0.1% for a given sample, present in at least 5% of samples at any abundance level.

Usage

```
TraditR2(X, Ab_min = 0.001)
```

Arguments

X	OTU COUNTS table, where taxa are columns and samples are rows of the table. It should be a in data frame format with columns corresponding to taxa names.
Ab_min	Numerical value, set to 0.001 by default. Throughout all samples, taxa with abundance less than this threshold with be removed.

Value

filtX	Filtered OTU table

Author(s)

Ekaterina Smirnova

References

Smirnova, E., Huzurbazar, H., Jafari, F. “PERFect: permutation filtration of microbiome data.

Examples

```
data(mock2)

# Counts data matrix
Counts <- mock2$Counts

# Filtering
Filtered_X <- TraditR2(Counts)
```

Index