Package 'metamicrobiomeR'

October 13, 2022

Title Microbiome Data Analysis & Meta-Analysis with GAMLSS-BEZI & Random Effects

Version 1.2

Description Generalized Additive Model for Location, Scale and Shape (GAMLSS) with zero inflated beta (BEZI) family for analysis of microbiome relative abundance data (with various options for data transformation/normalization to address compositional effects) and random effects meta-analysis models for meta-analysis pooling estimates across microbiome studies are implemented. Random Forest model to predict microbiome age based on relative abundances of shared bacterial genera with the Bangladesh data (Subramanian et al 2014), comparison of multiple diversity indexes using linear/linear mixed effect models and some data display/visualization are also implemented. The reference paper is published by Ho NT, Li F, Wang S, Kuhn L (2019) <doi:10.1186/s12859-019-2744-2> .

Depends R (>= 4.0.0), gamlss

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports meta, lme4, gdata, plyr, dplyr, tidyr, ggplot2, gridExtra, lmerTest, matrixStats, zCompositions, compositions

Suggests testthat, RCurl, httr, repmis, jsonlite, knitr, rmarkdown, grDevices, gplots, magrittr, tools, foreign, mgcv, reshape2, caret, randomForest, tsibble, RColorBrewer

VignetteBuilder knitr

URL https://github.com/nhanhocu/metamicrobiomeR

BugReports https://github.com/nhanhocu/metamicrobiomeR/issues

NeedsCompilation no

Author Nhan Ho [aut, cre]

Maintainer Nhan Ho <nhanhocumc@gmail.com>

Repository CRAN

Date/Publication 2020-11-09 11:20:05 UTC

R topics documented:

`alpha.compare` *Compare multiple alpha diversity indexes between groups*

Description

This function calculates average of alpha diversity indexes for a specific rarefaction depth, standardize diversity indexes and compare between groups using linear/linear mixed effect model and adjust for covariates.

Usage

```
alpha.compare(
  datlist,
  depth,
  mapfile,
  mapsampleid,
  comvar,
  adjustvar,
  personid = "personid",
  longitudinal = "yes",
```

```
  age.limit = 1e+06,
  standardize = FALSE
)
```

Arguments

datlist	the list of your dataframe.
depth	the rarefaction depth of choice. Depth can be "max" (highest depth) or an order (number) of the depth in the list generated by alpha_rarefaction.py
mapfile	mapping file
mapsampleid	sample id in the mapping file
comvar	variable for comparison
adjustvar	variables that need to be adjusted in the model
personid	name of variable for person id. Default is "personid" (applicable when longitudinal="yes").
longitudinal	longitudinal data or one time data. Options are c("yes","no"). Default is "yes".
age.limit	age upper limit for included samples. Default is 1000000 months (~no upper limit).
standardize	standardization of diversity indexes before comparison or not. Default is FALSE.

Value

list of comparison result matrices and mean diversity of all indexes for each samples (with or without standardization)

Examples

```
data(alphadat)
data(covar.rm)
covar.rm$sampleid<-tolower(covar.rm$sampleid)
#comparison of standardized alpha diversity indexes between genders adjusting for
#breastfeeding and infant age at sample collection in infants <=6 months of age
alphacom<-alpha.compare(datlist=alphadat,depth=3,mapfile=covar.rm,
mapsampleid="sampleid", comvar="gender",adjustvar=c("age.sample","bf"),
longitudinal="yes", age.limit=6,standardize=TRUE)
alphacom$alphasum[,1:5]
```

---

alphadat *Alpha diversity data.*

Description

Alpha diversity data from alpha_rarefaction output from QIIME.

Usage

```
data(alphadat)
```

Format

A list of 4 dataframes for four indexes: Chao1, Observed_species, PD_whole_tree, Shannon.

Source

Gordon Lab

References

Subramanian et al. Nature. 2014 Jun 19; 510(7505): 417–421. (PubMed)

Examples

```
data(alphadat)
# Load covariate data
data(covar.rm)
covar.rm$sampleid<-tolower(covar.rm$sampleid)
#comparison of standardized alpha diversity indexes between genders adjusting for
#breastfeeding and infant age at sample collection in infants <=6 months of age
alphacom<-alpha.compare(datlist=alphadat,depth=3,mapfile=covar.rm,
mapsampleid="sampleid", comvar="gender",adjustvar=c("age.sample","bf"),
longitudinal="yes", age.limit=6,standardize=TRUE)
alphacom$alphasum[,1:5]
```

---

asum4 *Combined alpha diversity data for meta-analysis.*

Description

Result outputs of differential analysis of alpha diversity using linear or linear mixed effects model from "alpha.compare" function combined from 4 studies for meta-analysis. The comparison was between gender adjusted for age of infants at sample collection.

Usage

```
data(asum4)
```

Format

A dataframe with 16 rows and 18 variables.

Source

Gordon Lab

References

Subramanian et al. Nature. 2014 Jun 19; 510(7505): 417–421. (PubMed)

Bender et al. Sci Transl Med. 2016 Jul 27; 8(349): 349ra100. (PubMed)

Pannaraj et al. JAMA Pediatr. 2017;90095(7):647–54. (PubMed)

Thompson et al. Front Cell Infect Microbiol. 2015;5:3. (PubMed)

Examples

```
data(asum4)
```

`covar.rm` *Covariate data.*

Description

Monthly longitudinal clinical data of 50 infants from birth to 2 years of life.

Usage

```
data(covar.rm)
```

Format

A dataframe with 996 rows and 32 variables.

Source

Gordon Lab

References

Subramanian et al. Nature. 2014 Jun 19; 510(7505): 417–421. (PubMed)

Examples

```
data(covar.rm)
# Load KEGG pathway data
data(kegg.12)
# Comparison of pathway relative abundances for some first pathways of level 1 only
# and assuming crosssectional data (to save running time)
path1<-pathway.compare(pathtab=list(kegg.12[[1]][, 1:2]),
mapfile=covar.rm,sampleid="sampleid",pathsum="rel", stat.med="gamlss",
comvar="gender",adjustvar=c("age.sample","bf"), longitudinal="no",
p.adjust.method="fdr", percent.filter=0.05,relabund.filter=0.00005)
taxcomtab.show(taxcomtab=path1$l1, sumvar="path",tax.lev="l2",
tax.select="none", showvar="genderMale", p.adjust.method="fdr",p.cutoff=1)
```

gtab.3stud *Test datasets for microbiome age prediction.*

Description

A list of three datasets to be used as test datasets for microbiome age prediction using Random Forest model.

Usage

```
data(gtab.3stud)
```

Format

A list of three dataframes.

Source

Gordon Lab

References

Subramanian et al. Nature. 2014 Jun 19; 510(7505): 417–421. (PubMed)

Bender et al. Sci Transl Med. 2016 Jul 27; 8(349): 349ra100. (PubMed)

Pannaraj et al. JAMA Pediatr. 2017;90095(7):647–54. (PubMed)

Thompson et al. Front Cell Infect Microbiol. 2015;5:3. (PubMed)

Examples

```
data(gtab.3stud)
```

kegg.12 *Pathway abundance data.*

Description

KEGG pathway abundance data from PICRUSt analysis. This is monthly longitudinal data of 50 infants from birth to 2 years of life.

Usage

```
data(kegg.12)
```

Format

A list of 2 dataframes for level 1 and level 2 of KEGG pathways.

Source

Gordon Lab

References

Subramanian et al. Nature. 2014 Jun 19; 510(7505): 417–421. (PubMed)

Examples

```
data(kegg.12)
# Load covariate data
data(covar.rm)
# Comparison of pathway relative abundances for some first pathways of level 1 only
# and assuming crosssectional data (to save running time)
path1<-pathway.compare(pathtab=list(kegg.12[[1]][, 1:2]),
mapfile=covar.rm,sampleid="sampleid",pathsum="rel", stat.med="gamlss",
comvar="gender",adjustvar=c("age.sample","bf"), longitudinal="no",
p.adjust.method="fdr", percent.filter=0.05,relabund.filter=0.00005)
taxcomtab.show(taxcomtab=path1$l1, sumvar="path",tax.lev="l2",
tax.select="none", showvar="genderMale", p.adjust.method="fdr",p.cutoff=1)
```

---

meta.niceplot *Nice meta-analysis plots.*

Description

This function displays meta-analysis results of relative abundance as a nice combined heatmap and forest plot. More flexibility/options for plot will be added.

Usage

```
meta.niceplot(
  metadat,
  sumtype = "taxa",
  level = "main",
  p,
  p.adjust,
  phyla.col = "rainbow",
 phyla.select = c("actinobacteria", "bacteroidetes", "cyanobacteria", "firmicutes",
    "fusobacteria", "proteobacteria", "verrucomicrobia", ".thermi."),
 col.select = c("#dd1c77", "#31a354", "#91003f", "#d95f0e", "#636363", "#2ef0e7",
    "#862ef0", "#000"),
 est.break = c(-Inf, -1, -0.5, -0.1, 0, 0.1, 0.5, 1, Inf),
 est.break.label = c("<-1", "[-1,-0.5)", "[-0.5,-0.1)", "[-0.1,0)", "[0,0.1)",
    "[0.1,0.5)", "[0.5,1)", ">=1"),
 neg.palette = "PuBu",
 pos.palette = "YlOrRd",
 p.sig.heat = "no",
```

```
  p.break = c(0, 1e-04, 0.05, 1),
  p.break.label = c("**", "*", ""),
  p.pool.break = c(0, 0.05, 1),
  p.pool.break.label = c("[0,0.05)", "[0.05,1]"),
  padjust.pool.break = c(0, 0.1, 1),
  padjust.pool.break.label = c("[0,0.1)", "[0.1,1]"),
  forest.est.shape = c("17", "16"),
  forest.est.col = c("red", "black"),
  forest.col = "by.pvalue",
  leg.key.size = 1,
  leg.text.size = 8,
  heat.text.x.size = 8,
  heat.text.x.angle = 0,
  forest.axis.text.y = 8,
  forest.axis.text.x = 8,
  heat.forest.width.ratio = c(1, 1),
  point.ratio = c(3, 1),
  line.ratio = c(2, 1)
)
```

Arguments

metadat	output data from metatab.show.
sumtype	Either "taxa" for taxa and "path" for pathway.
level	"main" for main level such as phylum or "sub" for higher level such as species. Default is "main".
p	name of variable for p-values
p.adjust	name of variable for multiple testing adjusted p-values
phyla.col	type of color for main level (phylum). Options are "rainbow" (default) or "select".
phyla.select	selected phyla for selected colors (only when phyla.col="select"). Default are c("actinobacteria","bacteroidetes","cyanobacteria","firmicutes","fusobacteria","proteobacteria","verrucor
col.select	selected colors for selected phyla (only when phyla.col="select"). Corresponding default are c("#dd1c77","#31a354","#91003f","#d95f0e","#636363","#2ef0e7","#862ef0","#000").
est.break	breaks for estimates to generate color categories on heatmap. Default are c(-Inf, -1,-0.5,-0.1,0,0.1,0.5,1, Inf). For pathway, recommended breaks are c(-Inf, -0.5,-0.1,-0.05,0,0.05,0.1,0.5, Inf).
est.break.label	labels for corresponding color categories on heatmap generated by est.break. Default corresponding to default est.break are c("<-1","[-1,-0.5)","[-0.5,-0.1)","[-0.1,0)", "[0,0.1)", "[01,0.5)", "[0.5,1)", ">=1"). For pathway, corresponding recommended labels are c("<-0.5)", "[-0.5,-0.1)","[-0.1,-0.05)","[-0.05,0)","[0,0.05)","[0.05,0.1)", "[0.1,0.5)", ">=0.5").
neg.palette	color palette for negative estimate values. Default is "PuBu". Use display.brewer.all() of RColorBrewer package for other options.

`pos.palette` color palette for positive estimate values. Default is "YlOrRd". Use display.brewer.all() of RColorBrewer package for other options.

`p.sig.heat` whether or not show significant p values on heatmap. Default is "yes".

`p.break` breaks for significant levels of p values. Default is c(0, 0.0001,0.05, 1).

`p.break.label` labels to be showed on heatmap for different levels of p-values from p.break. Default is c("**", "*","") for p breaks at c(0, 0.0001,0.05, 1).

`p.pool.break` breaks for pooled p-values to be distinguished in forest plot. Default are c(0,0.05,1).

`p.pool.break.label`
labels for pooled p-value breaks. Corresponding default are c("[0,0.05)","[0.05,1]").

`padjust.pool.break`
breaks for multiple testing adjusted p-values to be distinguished in forest plot. Default are c(0,0.1,1).

`padjust.pool.break.label`
labels for multiple testing adjusted p-value breaks. Corresponding default are c("[0,0.1)","[0.1,1]").

`forest.est.shape`
point shape of pooled estimates in forest plot. Default are c("17","16") for corresponding significant and non-significant pooled estimates.

`forest.est.col` colors of point (pooled estimates) and 95 CI bars in forest plot. Default are c("red", "black") for significant and non-significant estimates.

`forest.col` Color of forest plot (point estimates and 95 CI). Options are "by.pvalue" (distinguished by signficant vs. non-significant p-value) or "by.estimate" (color scaled similarly to heatmap color).

`leg.key.size` legdend key size for heatmap.

`leg.text.size` legend text size for heatmap.

`heat.text.x.size`
heatmap x label text size.

`heat.text.x.angle`
heatmap x label text angle.

`forest.axis.text.y`
forest plot y label text size.

`forest.axis.text.x`
forest plot x label text size.

`heat.forest.width.ratio`
ratio of width between heatmap and forest plot to be used in grid.arrange. Dedault is c(1,1).

`point.ratio` ratio of point size between significant pooled estimate and non-significant pooled estimate. Default is c(3,1).

`line.ratio` ratio of error bar line size between significant pooled estimate and non-significant pooled estimate. Default is=c(2,1).

Value

combined heatmap forest plot.

Examples

```
# load saved GAMLSS-BEZI results of four studies
# for the comparison of bacterial taxa relative abundance
# between genders adjusted for breastfeeding and
# infant age at sample collection
data(tabsex4)
#select only taxonomies of a small phylum for meta-analysis example
# (to save running time)
tlm<-tabsex4$id[grep("k__bacteria.p__fusobacteria",tabsex4$id)]
# meta-analysis
metab.sex<-meta.taxa(taxcomdat=tabsex4[tabsex4$id %in% tlm,],
summary.measure="RR", pool.var="id", studylab="study", backtransform=FALSE,
percent.meta=0.5, p.adjust.method="fdr")
#show results by table and plot
#phylum
#table
metatab.show(metatab=metab.sex$random,com.pooled.tab=tabsex4[tabsex4$id %in% tlm,],
tax.lev="l2",showvar="genderMale",p.cutoff.type="p", p.cutoff=1,display="table")
#plot
metadat<-metatab.show(metatab=metab.sex$random, com.pooled.tab=tabsex4[tabsex4$id %in% tlm,],
tax.lev="l2", showvar="genderMale",p.cutoff.type="p", p.cutoff=1,display="data")
meta.niceplot(metadat=metadat,sumtype="taxa",level="main",
p="p",p.adjust="p.adjust", phyla.col="rainbow", p.sig.heat="yes",
heat.forest.width.ratio =c(1.5,1), leg.key.size=0.8, leg.text.size=10,
heat.text.x.size=10, heat.text.x.angle=0, forest.axis.text.y=8,
forest.axis.text.x=10, point.ratio = c(4,2),line.ratio = c(2,1))
```

---

meta.taxa *Meta-analysis of taxa/pathway abundance comparison.*

Description

This function does meta-analysis based on estimates and standard errors from taxa/pathway abundance comparison using random effect and fixed effect meta-analysis models.

Usage

```
meta.taxa(
  taxcomdat,
  estimate.pattern = "Estimate.",
  se.pattern = "Std. Error.",
  summary.measure = "RR",
  pool.var = "id",
  studylab = "study",
  backtransform = FALSE,
  percent.meta = 0.5,
  p.adjust.method = "fdr"
)
```

Arguments

taxcomdat	matrice of estimates and SE of all taxa/pathways combined from all included studies.
estimate.pattern	string pattern for estimates. Default is "Estimate.".
se.pattern	string pattern for standard error. Default is "Std. Error.".
summary.measure	"RR" for estimates from GAMLSS with BEZI family and "RD" for estimates from Linear/linear mixed effect model. Default is "RR"
pool.var	name of id variable for meta-analysis. Default is "id".
studylab	name of variable characterizing included studies. Default is "study".
backtransform	whether or not to perform backtransformation of the estimates. Default is FALSE.
percent.meta	the threshold percentage of number of studies that a taxa is available to do meta-analysis. Default is 0.5
p.adjust.method	method for multiple testing adjustment (available methods of the function p.adjust). Default is "fdr".

Value

a list of matrices of results for all variables in the comparison models.

Examples

```
# load saved GAMLSS-BEZI results of four studies
# for the comparison of bacterial taxa relative abundance between
# genders adjusted for breastfeeding and infant age at sample collection
data(tabsex4)
#select only taxonomies of a small phylum for meta-analysis example
# (to save running time)
tlm<-tabsex4$id[grep("k__bacteria.p__fusobacteria",tabsex4$id)]
# meta-analysis
metab.sex<-meta.taxa(taxcomdat=tabsex4[tabsex4$id %in% tlm,],
summary.measure="RR", pool.var="id", studylab="study",
backtransform=FALSE, percent.meta=0.5, p.adjust.method="fdr")
#show results by table and plot
#phylum
#table
metatab.show(metatab=metab.sex$random,com.pooled.tab=tabsex4[tabsex4$id %in% tlm,],
tax.lev="l2",showvar="genderMale",p.cutoff.type="p", p.cutoff=1,display="table")
```

`metatab.show` *Display meta-analysis results.*

Description

This function displays meta-analysis results of relative abundance as heatmap, forest plot, table or data.

Usage

```
metatab.show(
  metatab,
  com.pooled.tab,
  sumvar = "taxa",
  highest.lev = "g",
  tax.lev = "l2",
  showvar,
  estimate.pattern = "Estimate.",
  se.pattern = "Std. Error.",
  p.pattern = "Pr(>|t|)",
  readjust.p = FALSE,
  p.cutoff.type = "p",
  p.cutoff = 0.05,
  display = "plot",
  plot = "heatmap",
  fill.value = "log(OR)",
  grid = FALSE,
  digit = 2,
  p.digit = 4
)
```

Arguments

`metatab`	matrice of taxa/pathway abundance comparison meta-analysis results generated from meta.taxa.
`com.pooled.tab`	matrice of taxa/pathway abundance comparison generated from taxa.compare or pathway.compare combined from all included studies.
`sumvar`	Either "taxa" for taxa and "path" for pathway.
`highest.lev`	Highest level of bacterial taxonomies available for analysis. Options are "g" for genus (usually for 16S data) and "s" for species (usually for shortgun data).
`tax.lev`	taxa level to be displayed. Options are from "l2" (phylum) to "l7" (species). Default is "l2".
`showvar`	variable (string pattern) in the model to be displayed.
`estimate.pattern`	string pattern for estimates. Default is "Estimate.".

se.pattern	string pattern for standard error variable. Default is "Std. Error.".
p.pattern	string pattern for p-value variable. Default is "Pr(>|t|)".
readjust.p	multiple testing re-adjustment for only the level to be displayed (TRUE) or keep original multiple testing adjustment for all taxa of all levels (FALSE).Default is FALSE.
p.cutoff.type	type of p-value for cutoff. Options are "p" for p-value or "p.adjust" for multiple testing adjusted p-value. Default is "p".
p.cutoff	cutoff p-value to be displayed. Default is 0.05.
display	type of display. Options are display=c("plot","table","data")
plot	type of plot. Options are plot=c("heatmap","forest").
fill.value	name of legend.
grid	whether multiple plots will be displayed alongside. Default is FALSE.
digit	digit for estimates and 95 CI. Default is 2.
p.digit	digit for p-values. Default is 4.

Value

plot table or data.

Examples

```
# Load saved GAMLSS-BEZI results of four studies for the comparison of
# bacterial taxa relative abundance between genders adjusted for
# breastfeeding and infant age at sample collection
data(tabsex4)
#select only taxonomies of a small phylum for meta-analysis example
# (to save running time)
tlm<-tabsex4$id[grep("k__bacteria.p__fusobacteria",tabsex4$id)]
# meta-analysis
metab.sex<-meta.taxa(taxcomdat=tabsex4[tabsex4$id %in% tlm,],
summary.measure="RR", pool.var="id", studylab="study",
backtransform=FALSE, percent.meta=0.5, p.adjust.method="fdr")
#show results by table and plot
#phylum
#table
metatab.show(metatab=metab.sex$random,com.pooled.tab=tabsex4[tabsex4$id %in% tlm,],
tax.lev="l2",showvar="genderMale",p.cutoff.type="p", p.cutoff=1,display="table")
#plot
metadat<-metatab.show(metatab=metab.sex$random,com.pooled.tab=tabsex4[tabsex4$id %in% tlm,],
tax.lev="l2",showvar="genderMale",p.cutoff.type="p", p.cutoff=1,display="data")
meta.niceplot(metadat=metadat,sumtype="taxa",level="main",p="p",
p.adjust="p.adjust",phyla.col="rainbow",p.sig.heat="yes",
heat.forest.width.ratio =c(1.5,1), leg.key.size=0.8,
leg.text.size=10, heat.text.x.size=10, heat.text.x.angle=0,
forest.axis.text.y=8,forest.axis.text.x=10,
point.ratio = c(4,2),line.ratio = c(2,1))
```

`microbiomeage` *Predict microbiome age.*

Description

This function predicts microbiome age using Random Forest model based on relative abundances of bacterial genera shared with the Bangladesh study (Subramanian et al 2014). This function gets the shared genera list between the Bangladesh study and all other included studies, get the training and test sets from Bangladesh data based on the shared genera list, fit the train Random Forest model and predict microbiome age in the test set of Bangladesh data and data from all included studies, check for performance of the model based on the shared genera list on Bangladesh healthy cohort data, reproduce the findings of the Bangladesh malnutrition study.

Usage

```
microbiomeage(l6.relabundtab, bal6)
```

Arguments

`l6.relabundtab`	list of taxa summary table from phylum up to genus level merged to mapping file outputed from QIIME of all included studies.
`bal6`	reference data for model training (taxa summary table from phylum up to genus level merged to mapping file outputed from QIIME of the Bangladesh study).

Value

list of training and test sets of Bangladesh data, shared genera list, relative abundance data of shared genera, randomforest fit, RF model performance plot,predicted microbiome age of Bangladesh data and data of other included studies.

Examples

```
# The data used for this example are available
# in the "metamicrobiomeR" package version in Github.
# Download example data from the package github repo
#setwd("your directory") #put your working directory inside the quotation marks
download.file(url = "https://github.com/nhanhocu/metamicrobiomeR/archive/master.zip",
destfile = "metamicrobiomeR-master.zip")
# unzip the .zip file
unzip(zipfile = "metamicrobiomeR-master.zip")
#Load data from each study and put in a list
#Load Bangladesh train data
patht<-paste(getwd(),
"metamicrobiomeR-master/inst/extdata/QIIME_outputs/Bangladesh/tax_mapping7", sep="/")
bal6 <- utils::read.delim(paste(patht, "Subramanian_et_al_mapping_file_L6.txt", sep="/"))
colnames(bal6)<-tolower(colnames(bal6))
# Load data of 3 other studies
```

```
#format for data of other studies should be similar to Bangladesh data,
# must have 'age.sample' variable as age of infant at stool sample collection
data(gtab.3stud)
names(gtab.3stud)
#predict microbiome age on Bangladesh data and
# data of other three studies based on shared genera across 4 studies
#Predict microbiome age on train and test data (take time to run)
miage<-microbiomeage(l6.relabundtab=gtab.3stud, bal6=bal6)
#list of shared genera that are available in the Bangladesh study
# and other included studies
miage$sharedgenera.importance
#check performance
gridExtra::grid.arrange(miage$performanceplot$ptrain, miage$performanceplot$ptest,nrow=1)
#replicate the findings of Subramanian et al paper
ggplot2::ggplot() +geom_point(data=miage$microbiomeage.bangladesh$all,
aes(x=age.sample, y=age.predicted, colour=health_analysis_groups))
```

`pathway.compare` *Compare (kegg) pathway abundance*

Description

This is a slightly modified version of the taxa.compare function. It compares pathway abundance generated from PICRUSt analysis between groups using different methods (apply to pathway summary tables already merged to mapping file and put in a list (e.g.level 1, 2 and 3)). Specifically, it compares relative abundances of bacterial functional pathways at all levels using GAMLSS or LM/LMEM and compares of log(absolute abundances) of bacterial functional pathways at all levels using LM/LMEM.

Usage

```
pathway.compare(
  pathtab,
  mapfile,
  sampleid = "sampleid",
  pathsum = "rel",
  stat.med = "gamlss",
  transform = "none",
  comvar,
  adjustvar,
  personid = "personid",
  longitudinal = "yes",
  p.adjust.method = "fdr",
  percent.filter = 0.05,
  relabund.filter = 5e-05,
  pooldata = FALSE
)
```

Arguments

pathtab	list of pathway abundance table of all levels.
mapfile	mapping file or file containing covariates.
sampleid	variable containing sample id to be matched between pathway abundance table and mapping file.
pathsum	type of abundance to be compared. Options are "rel" for relative abundance or "log" for log of absolute abundance. Default is "rel".
stat.med	statistical method for comparison. stat.med can be "lm" for LM/LMEM (usable for both pathsum="rel" or "log") or "gamlss" for GAMLSS with BEZI family (gamlss only make sense if pathsum="rel").
transform	transformation of relative abundance data. Options are "none" for no transformation, "asin.sqrt" for arcsine transformation, "logit" for logit transformation. Default is "none".
comvar	main variable for comparison.
adjustvar	variables to be adjusted.
personid	name of variable for person id in mapping file (applicable for longitudinal data)
longitudinal	whether the data is longitudinal. Default is "yes".
p.adjust.method	method for multiple testing adjustment. Available options are those of the p.adjust function. Default is "fdr".
percent.filter	prevalence threshold (the percentage of number of samples the taxa/pathway available). Default is 0.05.
relabund.filter	relative abundance threshold (the minimum of the average relative abundance for a taxa/pathway to be retained). Default is 0.00005.
pooldata	whether the data is pooled from multiple studies. Default is FALSE.

Value

matrice of coefficients, standard errors, p-values and multiple testing adjusted p-values of all variables in the models.

Examples

```
#Load Bangladesh pathway and metadata
data(kegg.12)
data(covar.rm)
# Comparison of pathway relative abundances for some first pathways of level 1 only
# and assuming crosssectional data (to save running time)
path1<-pathway.compare(pathtab=list(kegg.12[[1]][, 1:2]),
mapfile=covar.rm,sampleid="sampleid",pathsum="rel", stat.med="gamlss",
comvar="gender",adjustvar=c("age.sample","bf"), longitudinal="no",
p.adjust.method="fdr", percent.filter=0.05,relabund.filter=0.00005)
taxcomtab.show(taxcomtab=path1$l1, sumvar="path",tax.lev="l2",
tax.select="none", showvar="genderMale", p.adjust.method="fdr",p.cutoff=1)
```

read.multi *Read multiple files*

Description

This function reads multiple files of the same pattern in a directory into R.

Usage

```
read.multi(
  patht,
  patternt = ".txt",
  assignt = "no",
  study = NULL,
  tolower.colnames = TRUE
)
```

Arguments

patht	path to files.
patternt	pattern of files. Options are ".txt" or ".csv".
assignt	assign files. Default is "no".
study	name of the study. Default is NULL.
tolower.colnames	turn all column names to lower case. Default is TRUE.

Value

list of all data files in the path

Examples

```
# The data used for this example are available
# in the "metamicrobiomeR" package version in Github.
# Download example data from the package github repo
#setwd("your directory") #put your working directory inside the quotation marks
download.file(url = "https://github.com/nhanhocu/metamicrobiomeR/archive/master.zip",
destfile = "metamicrobiomeR-master.zip")
# unzip the .zip file
unzip(zipfile = "metamicrobiomeR-master.zip")
patht<-paste(getwd(),
"metamicrobiomeR-master/inst/extdata/QIIME_outputs/Bangladesh/alpha_div_collated", sep="/")
alpha.ba<-read.multi(patht=patht,patternt=".txt", assignt="no",study="Bangladesh")
```

tabsex4 *Combined data for meta-analysis.*

Description

Result outputs of differential abundance analysis using GAMLSS_BEZI from "taxa.compare" function combined from 4 studies for meta-analysis. The comparison was between gender adjusted for age of infants at sample collection.

Usage

```
data(tabsex4)
```

Format

A dataframe with 701 rows and 23 variables.

Source

Gordon Lab

References

Subramanian et al. Nature. 2014 Jun 19; 510(7505): 417–421. (PubMed)

Bender et al. Sci Transl Med. 2016 Jul 27; 8(349): 349ra100. (PubMed)

Pannaraj et al. JAMA Pediatr. 2017;90095(7):647–54. (PubMed)

Thompson et al. Front Cell Infect Microbiol. 2015;5:3. (PubMed)

Examples

```
# load saved GAMLSS-BEZI results of four studies
# for the comparison of bacterial taxa relative abundance between
# genders adjusted for breastfeeding and infant age at sample collection
data(tabsex4)
#select only taxonomies of a small phylum for meta-analysis example
# (to save running time)
tlm<-tabsex4$id[grep("k__bacteria.p__fusobacteria",tabsex4$id)]
# meta-analysis
metab.sex<-meta.taxa(taxcomdat=tabsex4[tabsex4$id %in% tlm,],
summary.measure="RR", pool.var="id", studylab="study",
backtransform=FALSE, percent.meta=0.5, p.adjust.method="fdr")
#show results by table and plot
#phylum table
metatab.show(metatab=metab.sex$random,com.pooled.tab=tabsex4[tabsex4$id %in% tlm,],
tax.lev="l2",showvar="genderMale",p.cutoff.type="p", p.cutoff=1,display="table")
```

taxa.compare *Compare taxa relative abundance*

Description

This function compares taxa relative abundance summary tables at all levels between groups using GAMLSS with BEZI or Linear/Linear Mixed Effect models (LM/LMEM) after filtering (using prevalence and relative abundance thresholds).

Usage

```
taxa.compare(
  taxtab,
  propmed.rel = "gamlss",
  transform = "none",
  zeroreplace.method = "none",
  comvar,
  adjustvar,
  personid = "personid",
  longitudinal = "yes",
  percent.filter = 0.05,
  relabund.filter = 5e-05,
  p.adjust.method = "fdr"
)
```

Arguments

taxtab	taxa relative abundance table (already merged to mapping file) from phylum to species or any preferred highest taxa level.
propmed.rel	statistical method for comparing relative abundance. Options are "lm" for LM/LMEM or "gamlss" for GAMLSS with BEZI family.
transform	transformation of relative abundance data. Options are "none" for no transformation, "gmpr" for Geometric Mean of Pairwise Ratios (GMPR) normalization, "asin.sqrt" for arcsine transformation, "logit" for logit transformation, "clr" for centered log ratio transformation. Default is "none".
zeroreplace.method	Method for zero replacement implemented in R package *zCompositions*. Options are "none" for no replacement, "multKM" for Multiplicative Kaplan-Meier smoothing spline (KMSS) replacement, "multLN" for Multiplicative lognormal replacement, "multRepl" for Multiplicative simple replacement, "lrEM" for Log-ratio EM algorithm, "lrDA" for Log-ratio DA algorithm. Default is "none".
comvar	main variable for comparison
adjustvar	variables to be adjusted.
personid	name of variable for person id (applicable for longitudinal data)
longitudinal	whether data is longitudinal? Options are "yes" or "no". Default is "yes".

`percent.filter` prevalence threshold (the percentage of number of samples the taxa/pathway available). Default is 0.05.

`relabund.filter` relative abundance threshold (the minimum of the average relative abundance for a taxa/pathway to be retained). Default is 0.00005.

`p.adjust.method` method for multiple testing adjustment. Options are those of the p.adjust.methods of stats:: p.adjust function. Default for this function is "fdr".

Value

matrice of coefficients, standard errors, p-values and multiple testing adjusted p-values of all variables in the models.

Examples

```
#Load summary tables of bacterial taxa relative abundance from Bangladesh data
data(taxtab6)
tab6<-as.data.frame(taxtab6)
tl<-colnames(taxtab6)[grep("k__bacteria.p__fusobacteria",colnames(taxtab6))]
taxacom.ex<-taxa.compare(taxtab=tab6[,c("personid","x.sampleid","bf","age.sample",tl)],
propmed.rel="gamlss",comvar="bf",adjustvar="age.sample",
longitudinal="yes",p.adjust.method="fdr")
```

---

taxa.filter *Filter relative abundance data*

Description

This function filters bacterial taxa or pathway relative abundance tables based on the percentage of samples with their availability (prevalence) and relative abundance thresholds. It will remove taxa/pathway with relative abundance <relabund.filter and available in <percent.filter of number of samples.

Usage

```
taxa.filter(taxtab, percent.filter = 0.05, relabund.filter = 5e-05)
```

Arguments

`taxtab` taxa/pathway relative abundance table.

`percent.filter` prevalence threshold (the percentage of number of samples the taxa/pathway available). Default is 0.05.

`relabund.filter` relative abundance threshold (the minimum of the average relative abundance for a taxa/pathway to be retained). Default is 0.00005.

Value

list of all taxa/pathways retained after filtering.

Examples

```
#Load summary tables of bacterial taxa relative abundance from Bangladesh data
data(taxtab6)
taxlist.rm<-taxa.filter(taxtab=taxtab6,percent.filter = 0.05,
relabund.filter = 0.00005)
```

---

taxa.mean.plot *Plot mean taxa abundance*

Description

This function visualize mean relative abundance by group as stacked plots.

Usage

```
taxa.mean.plot(
  tabmean,
  sumvar = "taxa",
  tax.select = "none",
  tax.lev = "l2",
  comvar,
  groupvar,
  mean.filter = 0.005,
  pallete.by.phylum = FALSE,
  show.taxname = "full",
  legend.position = "right",
  xlab = "",
  ylab = "Relative abundance"
)
```

Arguments

tabmean	table of mean abundance generated from taxa.meansdn.
sumvar	variable to be plotted. Options are c("taxa","path"). Default is "taxa"
tax.select	list of selected taxa/pathways to be plotted. Default is "none" or plot all taxa/pathways.
tax.lev	taxa level to be visualized. Options are from "l2" (phylum) to "l7" (species). Default is "l2". If sumvar="path", all pathways will be visualized.
comvar	main variable for comparison.
groupvar	variable for stratifying.
mean.filter	mean abundance filtering threshold (only plot those with mean abundance>threshold and plot all those with mean abundance <threshold as "other").

pallete.by.phylum
: whether each pallete of color for each phylum. Default is FALSE (plot distinc colors).

show.taxname
: whether show "full" taxa name or "short" name. Default is "full".

legend.position
: position of legend. Options are c("right", "left","bottom","top","none") as in ggplot2. Default is "right".

xlab
: label for x-axis. Default is "Chronological age (month)".

ylab
: label for y-axis. Default is "Relative abundance".

Value

a list of ggplot2 object and list of taxa/pathways plotted (those with mean abundance >mean.filter).

Examples

```
#Load summary tables of bacterial taxa relative abundance from Bangladesh data
data(taxtab6)
taxlist.rm<-taxa.filter(taxtab=taxtab6, percent.filter = 0.05, relabund.filter = 0.00005)
taxa.meansdn.rm<-taxa.meansdn(taxtab=taxtab6,sumvar="bf",groupvar="age.sample")
taxa.meansdn.rm<-taxa.meansdn.rm[taxa.meansdn.rm$bf!="No_BF",]
taxa.meansdn.rm$bf<-gdata::drop.levels(taxa.meansdn.rm$bf,reorder=FALSE)
#phylum
p.bf.l2<-taxa.mean.plot(tabmean=taxa.meansdn.rm, tax.lev="l2",
comvar="bf", groupvar="age.sample",mean.filter=0.005, show.taxname="short")
p.bf.l2$p
```

---

taxa.meansdn *Summarize abundance by group*

Description

This function summarizes taxa/pathway abundance tables to provide mean, sd, count by groups.

Usage

```
taxa.meansdn(
  taxtab,
  sumvar,
  groupvar,
  percent.filter = 0.05,
  relabund.filter = 5e-05,
  othervar = "none"
)
```

Arguments

taxtab	taxa/pathway abundance table from phylum to species or any preferred highest taxa level.
sumvar	main variable for summary
groupvar	variable to be stratified.
percent.filter	prevalence threshold (the percentage of number of samples the taxa/pathway available). Default is 0.05.
relabund.filter	relative abundance threshold (the minimum of the average relative abundance for a taxa/pathway to be retained). Default is 0.00005.
othervar	vector of variables that are not abundance variables to be summarized. Default is "none".

Value

table of mean, sd, count by group.

Examples

```
#Load summary tables of bacterial taxa relative abundance from Bangladesh data
data(taxtab6)
taxa.meansdn.rm<-taxa.meansdn(taxtab=taxtab6,sumvar="bf",groupvar="age.sample")
```

---

taxcomtab.show *Display abundance comparison results.*

Description

This function displays taxa/pathway abundance comparison results as table.

Usage

```
taxcomtab.show(
  taxcomtab,
  sumvar = "taxa",
  tax.lev = "l2",
  tax.select = "none",
  showvar,
  readjust.p = FALSE,
  p.adjust.method = "fdr",
  p.cutoff = 0.05,
  digit = 2,
  p.digit = 4
)
```

Arguments

taxcomtab	table of taxa abundance comparison generated from taxa.compare.
sumvar	Options are "taxa" for bacterial taxa and "path" for pathway. Default is "taxa"
tax.lev	taxa level to be displayed. Options are from "l2" (phylum) to "l7" (species). Default is "l2".
tax.select	selected list of taxa to be displayed. Default is "none" or display all available taxa.
showvar	variable (pattern) in the model to be displayed.
readjust.p	multiple testing re-adjustment for only the level to be displayed (TRUE) or keep original multiple testing adjustment for all taxa of all levels (FALSE).
p.adjust.method	method for multiple testing adjustment. Available options are those of the p.adjust function. Default is "fdr".
p.cutoff	cutoff p-value to be displayed. Default is 0.05.
digit	digit for estimates and 95 CI. Default is 2.
p.digit	digit for p-values. Default is 4.

Value

a table of results.

Examples

```
#Load summary tables of bacterial taxa relative abundance from Bangladesh data
data(taxtab6)
tab6<-as.data.frame(taxtab6)
#Comparison of bacterial taxa relative abundance using GAMLSS
# Only run on a few taxa of a phylum to save running time
tl<-colnames(taxtab6)[grep("k__bacteria.p__fusobacteria",colnames(taxtab6))]
taxacom.ex<-taxa.compare(taxtab=tab6[,c("personid","x.sampleid","bf","age.sample",tl)],
propmed.rel="gamlss",comvar="bf",adjustvar="age.sample",
longitudinal="yes",p.adjust.method="fdr")
# show phylum results
taxcomtab.show(taxcomtab=taxacom.ex,tax.select="none",
showvar="bfNon_exclusiveBF", tax.lev="l2",
readjust.p=TRUE,p.adjust.method="fdr",p.cutoff = 1)
```

---

taxtab6 *Taxonomic relative abundance data.*

Description

Monthly longitudinal relative abundance data from phylum to genus level of 50 infants from birth to 2 year of life. Mapping file is merged to the data for ready use.

Usage

```
data(taxtab6)
```

Format

A data frame with 322 row (samples) and 803 variables (including mapping varilable and bacterial taxonomies from phylum to genus level).

Source

Gordon Lab

References

Subramanian et al. Nature. 2014 Jun 19; 510(7505): 417–421. (PubMed)

Examples

```
#Load summary tables of bacterial taxa relative abundance from Bangladesh data
data(taxtab6)
tab6<-as.data.frame(taxtab6)
tl<-colnames(taxtab6)[grep("k__bacteria.p__fusobacteria",colnames(taxtab6))]
taxacom.ex<-taxa.compare(taxtab=tab6[,c("personid","x.sampleid","bf","age.sample",tl)],
propmed.rel="gamlss",comvar="bf",adjustvar="age.sample",
longitudinal="yes",p.adjust.method="fdr")
```

Index