Package 'MTA'

July 6, 2021

Title Microbial Trend Analysis for Common Dynamic Trend, Group
Comparison, and Classification in Longitudinal Microbiome Study

Version 0.1.0

Date 2021-06-23

Author Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

Maintainer
Chan Wang <Chan.Wang@nyulangone.org> and Huilin Li <Huilin.Li@nyulangone.org>

Description
Microbial trend analysis (MTA) framework is proposed for the high-dimensional and phyloge-
netic longitudinal microbiome data. The MTA framework can describe microbial dynamics, differ-
ential group difference, extract key taxa driving the microbial temporal trend, and classify the sub-
jects, is a highly in-time and attractive analytical tool to study the dynamics of microbiome com-
position and identify key bacterial species that affect susceptibility to complex diseases.

Depends R (>= 4.0.0)

Imports adephylo, fda, ggplot2, reshape2, toOrdinal, MASS, stats,
utils

License GPL (>= 2)

Encoding UTF-8

LazyData true

URL https://github.com/chanw0/MTA

BugReports http://github.com/chanw0/MTA/issues

RoxygenNote 7.1.1

R topics documented:

MTA-package	*Microbial Trend Analysis for Common Dynamic Trend, Group Comparison, and Classification in Longitudinal Microbiome Study*

Description

Microbial trend analysis (MTA) framework is an attractive and effective tool in investigating the dynamic microbial pattern in the longitudinal microbiome study. In particular, MTA can perform three tasks: 1) MTA can capture the common and major community level microbial dynamic trends for a given group of subjects and identify the dominant taxa; 2) MTA can examine whether the microbial overall dynamic trends are significantly different or not in groups; 3) MTA can classify a given subject based on its longitudinal microbial profiling. Both simulations and real data applications showed that MTA framework is robust and powerful in hypothesis testing, taxon identification, and subject classification.

Details

Package:	MTA
Type:	Package
Version:	1.0
Date:	2021-07-06
License:	GPL (>= 2)

Author(s)

Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

Maintainer: Chan Wang <Chan.Wang@nyulangone.org>, Huilin Li <Huilin.Li@nyulangone.org>

References

Wang C, Hu J, Blaser M J, Li H (2021). Microbial Trend Analysis for Common Dynamic Trend, Group Comparison, and Classification in Longitudinal Microbiome Study.

MTA	*The overall function in MTA framework*

Description

This overall function performs three tasks: 1) capturing the common and major community level microbial dynamic trends for a given group of subjects and identifying the dominant taxa; 2) examining whether the microbial overall dynamic trends are significantly different or not in groups; 3) classifying a given subject based on its longitudinal microbial profiling.

Usage

```
MTA(Y1,Y_new=NULL,phy.tree=NULL,timevec=NULL, transf="None",
           M=NULL,proportion.explained=0.85,
           k=5,lambda1.set=c(0.01,0.1,1,10),
           lambda2.set=seq(0,1,0.001),lambda3.set=c(0),num.sam=10,alpha=0.05)
```

Arguments

Y1	Y1 can be an N*P*T numeric array containing compositional microbiome data for N subjects having P taxa across T time points, or a list, in which each element represents a group of subjects having P taxa across T time points. The number of groups is equal to or larger than 2. The list's name can be assigned as the group labels, otherwise, it is 1:length(Y1) by default. If Y1 is an N*P*T numeric array, MTA gives the microbial dynamics for Y1 and identifies the key taxa. If Y1 is a list, MTA provides the group comparison or classification.
Y_new	An optional N*P*T numeric array containing compositional microbiome data for N subjects having P taxa across T time points. Y_new represents the subjects who need to be classified based on the training set Y1. Default=NULL, no calculation for the classification part.
phy.tree	The phylogenetic tree (phylo-class) object for P taxa. Default=NULL, no Laplacian penalty dealing with the phylogenetical correlation among taxa.
timevec	An optional numeric vector of the continuous time point with length = T.
transf	A character value, which represents the transformation technique one wants to take. "None" represents MTA analyzes the microbial relative abundance directly; "Log", "ALR", and "CLR" represent the corresponding log, additive log-ratio transformation and centered log-ratio transformation respectively. Default="None".
M	The number of top common trends one wants to select. Default=NULL.
proportion.explained	
	A numeric value used to determine the number of top common trends as an alternative. Default=0.85.
k	A numeric value represents k-fold cross-validation to determine the tuning parameters. Default=5.
lambda1.set	A numeric vector containing the candidate tunning parameters for the smoothing technique to deal with the smoothness of the extracted trends. Default=c(0.01,0.1,1,10).
lambda2.set	A numeric vector containing the candidate tunning parameters for the Lasso penalty to deal with the high dimensionality of taxa. Default=seq(0,1,0.001).
lambda3.set	A numeric vector containing the candidate tunning parameters for the Laplacian penalty to deal with the phylogenetical correlation. Default=c(0).
num.sam	An integer value, the number of resampling for the group comparison. Default=10.
alpha	An numeric value, the signficance level for the group comparison. Default=0.05.

Value

When Y1 is an N*P*T numeric array, MTA performs the first task: capturing the common and major community level microbial dynamic trends and identifying the dominant taxa based on the Y1. A list which contains two elements:

Trend Plot The extracted common trends extracted from all subjects Y1

Factor Score The estimated factor scores for all key taxa which contribute to the extracted common trends.

When Y1 is a list involving two or more than two groups, but Y_new is NULL, MTA performs the second task: examining whether the microbial overall dynamic trends are significantly different or not in groups and provides a list which contains comparing results between any two groups. The comparing results between any two groups contains five elements:

P value P value for the common trends representing the difference in the microbial dynamic pattern between these two groups

Trends The extracted common trends among R resamplings

Discover rate The discover rates for all taxa among R resamplings. This item is provided when the common trend is significant, otherwise, on this item.

Factor Score The estimated factor scores for all taxa which contribute to the common trends. This item is provided when the common trend is significant, otherwise, on this item.

Standard error The standard error of the estimated factor scores for all taxa among R resamplings. This item is provided when the common trend is significant, otherwise, on this item.

When Y1 is a list involving two or more than two groups, and Y_new is an N*P*T numeric array, MTA performs the third task: classifying the subjects in Y_new based on the training set Y1. The result contains an N*2 matrix, in which the first column is the subject's ID and the second column is its group label accordingly.

Author(s)

Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

Maintainer: Chan Wang <Chan.Wang@nyulangone.org>, Huilin Li <Huilin.Li@nyulangone.org>

References

Wang C, Hu J, Blaser M J, Li H (2021). Microbial Trend Analysis for Common Dynamic Trend, Group Comparison, and Classification in Longitudinal Microbiome Study.

Examples

```
########### generation data
library(dirmult)
## sample size
N=20;
## time point
T=10;
##number of taxa
P=20;

beta0=abs(rnorm(P,10,20))
## key taxa
causal.ind=sample(1:P,3)

#### generating two groups
group1=array(NA, dim = c(N,P,T))
for(t in 1:T)  group1[, ,t] = rdirichlet(N,beta0)
```

```
group2=array(NA, dim = c(N,P,T))
for(t in 1:T){
  betaT=rep(0,P)
  betaT[causal.ind]=c(-1,2, -1)/2*min(beta0[causal.ind])*sin(t)
  group2[, ,t] = rdirichlet(N,(beta0+betaT))}


########## MTA captures the common trends shared by all subjects in group1
### extracting the common trend from a group of subjects with explained variation >=85%
MTA(Y1=group1)

### extracting the common trend from a group of subjects with 2 common trends
MTA(Y1=group1,M=2)


######## comparing between group1 and group2
MTA(list(group1,group2))


######## classifying subjects based on the training set
MTA(list(group1,group2),group2[1:5,,])
```

Toydata	*Toy data used for illustration*

Description

Toydata is a list with the length=3.

Usage

```
Toydata
```

Format

A list with the length=3 with names Case, Control, and phy.tree:

Case A 15 * 35 * 4 Array of relative abundances for 15 Cases

Control A 15 * 35 * 4 Array of relative abundances for 15 Controls

phy.tree The phylogenetic tree (phylo-class) object for 35 taxa

Examples

```
####### Toy data for illustration
library(ggpubr)
library(toOrdinal)
library(reshape2)

Case=Toydata[[1]]
Control=Toydata[[2]]
phy.tree=Toydata[[3]]
```

```
### Group comparison

aa=MTA(list(Control,Case),phy.tree=phy.tree)$`Comparison between group 1 and group 2`


aa1=aa[[4]]
aa2=aa[[5]]

for(i in 1:nrow(aa2)) {

 bb1=aa1[i,-1];bb2=aa2[i,-1]
 bb=order(bb1^2/sum(bb1^2),decreasing =TRUE)
 bb1[-c(bb[1:which(cumsum(cumsum(sort(bb1^2/sum(bb1^2),decreasing =TRUE))>0.99)==1)])]=0

 bb1[((bb1+1.96*bb2)>=0 & (bb1-1.96*bb2)<=0)]=NA

 aa1[i,-1]=bb1
}
effect1=aa1
colnames(effect1)=c("trend",unlist(dimnames(Control)[2]))
effect1=data.frame(effect1,check.names =FALSE)
effect1=melt(effect1,id.vars = 1,variable.name = "taxa",value.name = "score")
effect1$trend=paste(sapply(effect1$trend, toOrdinal), "common trend")

effect1=na.omit(effect1)

pp1=ggbarplot(effect1, x = "taxa", y = "score",
              ylab = "Estimated factor score", xlab="", facet.by = "trend",
              add = "mean")+
 facet_wrap(~trend,scales="free")+theme(axis.text.x = element_text(angle = 90, hjust = 1))

######### trend figure

trend1=aa[[2]]

pp2=ggline(trend1, x = "time",y = "Escore",
         ylab = "Common trend", xlab="Week", facet.by = "trend",
         add = "mean")+
 facet_wrap(~trend,scales="free_y")


ggarrange(pp2,pp1,labels = c("(A)", "(B)"), ncol = 2)

#### Classification
aa=MTA(list(Control,Case),Case, phy.tree=phy.tree)

### Group label 1: the first group (Control); 2: the second group (Case)
print(aa)
```

Index