

Package ‘AMDA’

January 2, 2019

Type Package

Title Adaptive Microbiome Differential Analysis (AMDA)

Version 1.0

Date 2019-01-02

Author Xiang Zhan

Maintainer Xiang Zhan <xiangzhan9@gmail.com>

Description This R package AMDA introduces an adaptive procedure to examine microbiome differential abundance under two different conditions.

License GPL (>=2)

R topics documented:

AMDA 1

AMDA	<i>Adaptive Microbiome Differential Analysis</i>
------	--

Description

This function adaptively examines microbiome differential abundances of a given taxa-set under two different conditions.

Usage

```
AMDA(X, g2, B = 1000)
```

Arguments

X	A n by p microbiome compositional matrix, with each row being a microbiome sample (relative abundances sum to 100%) and each column being a taxon.
g2	A n by 1 numerical vector taking values 1 and 2 denoting two different groups. The order of g2 matches the order of rows of X.
B	The number of permutations to calculate the p-value. The default value is 1000.

Details

To examine differential abundances of a taxa-set under two different conditions, the AMDA procedure first selects a subset of taxa that are more likely to be differentially abundant between two conditions as the testing subset. Then, permutations are used to establish the statistical significance of the differential abundances of the selected taxa-set.

Value

A p-value

References

Banerjee, K. et al. (2019) An adaptive multivariate two-sample test with application to microbiome differential abundance analysis. Submitted to *Frontiers in Genetics*.

Examples

```
library(MASS)
n1=n2=25
n=n1+n2
g2=c(rep(1,n1),rep(2,n2))
p=20
mu1=mu2=runif(p,min=0,max=10)
D=diag(sqrt(runif(p,min=1,max=3)),p)
A = matrix(0,p,p)
for(i in 1:p){
  for(j in 1:p){
    if (i==j) {A[i,j]=1}
    if(abs(i-j)==1) {A[i,j]=-0.5}
  }
}
sigma=D
sigma2=sigma1=sigma
W= mvrnorm(n1, mu1, sigma1, tol = 1e-6)
W= exp(W)
rowW = rowSums(W)
colW = colSums(W)
X1 = W
for(i in 1:nrow(W)){
  for(j in 1:ncol(W)){
    X1[i,j] = W[i,j]/rowW[i]
  }
}
W= mvrnorm(n2, mu2, sigma2, tol = 1e-6)
W= exp(W)
rowW = rowSums(W)
colW = colSums(W)
X2 = W
for(i in 1:nrow(W)){
  for(j in 1:ncol(W)){
    X2[i,j] = W[i,j]/rowW[i]
  }
}
X=rbind(X1,X2)
AMDA(X, g2, B=100)
```