Quick start guide for 'LRTT' package

Chao ZHOU [1,2], Tao WANG [1,2]

[1] Department of Bioinformatics and Biostatistics, Shanghai Jiao Tong University,
[2] SJTU-Yale Joint Center for Biostatistics, Shanghai Jiao Tong University,
Shanghai, China

March 27, 2018

1 Overview

R package 'LRTT' provides function to do differential abundance analysis for microbiome data incroporate phylogeny. This vignette give an introduction to the 'LRTT' package.

OTU table have to corresponds to the tree structure. We introduce four different simulation methods firstly, you can try any one you like. Then we can do the test based on the tree structure and the OTU table.The package can be installed and loaded with the command:

```
> # install.packages("devtools")
> library(devtools)
> install_github("ZRChao/LRTT")
> library(help = LRTT)
```

The follow steps show you how to use this packages. If any problems or suggestions, please free free to contact Chao ZHOU at Supdream8@sjtu.edu.cn.

2 Implementation in 'LRTT'

2.1 Simulation data

In this section, we introduce four simulation situation methods. BIT.Sim and DTM.Sim is based on the tree structure and return the count data includes the leaves and internal nodes of the tree; LNM.Sim and ANCOM.Sim just correspond to the differential OTU and return the OTU table. But we have bulid the index matrix function Taxa.index to return the relationship between leaves and internal nodes, the rows of the matrix is the leaves (OTUs) and columns are internal nodes, value 1 of the matrix means there are connected by branches otherwise is 0. Use this matrix, we can multiple it to the OTU table to get the internal nodes count.

```
> library(LRTT)
> p <- 10
> tree <- Tree.Sim(p)
> #plot.phylo(tree, type = "phylogram")
> taxa.index <- Taxa.index(p, tree)
> taxa.index
```

```
     11 12 13 14 15 16 17 18 19
[1,]  1  1  1  1  0  0  0  0  0
[2,]  1  1  1  1  0  0  0  0  0
[3,]  1  1  1  0  0  0  0  0  0
[4,]  1  1  0  0  1  0  0  0  0
```

```
 [5,]  1  1  0  0  1  0  0  0  0
 [6,]  1  0  0  0  0  1  1  0  0
 [7,]  1  0  0  0  0  1  1  1  1
 [8,]  1  0  0  0  0  1  1  1  1
 [9,]  1  0  0  0  0  1  1  1  0
[10,]  1  0  0  0  0  1  0  0  0
```

To simulation, we have to give probability on each branch which must statisfy that sums equal to 1. So after this setting, we can calculate the probability multiple along the branch to get the probability for the leaves which also statisfy all the probablity on the leaves sums to 1, otherwise something wrong.

```
> set.seed(2)
> dif.taxa <- sample(tree$edge[, 1], 1)
> prob <- Prob.branch(tree, seed = 1, dif.taxa)
> prob.m1 <- Prob.mult(p, tree, prob[, 1])
> prob.m2 <- Prob.mult(p, tree, prob[, 2])
> dif.otu <- which(prob.m1 != prob.m2)
> dif.otu

[1] 1 2
```

There can be some degenetation on the leaves which is small probability things. With the above settings and parameters, we can do next four simulations.

```
> ## BIT and DTM will return all count data
> data.bit <- BIT.Sim(p, seed = 2, N = 20, tree = tree, prob[, 1], prob[, 2])
> # data.dtm <- DTM.Sim(p, seed = 2, N = 20, tree, prob[, 1], prob[, 2], theta = 0.1)
> dim(data.bit)

[1] 40 18

> # dim(data.dtm)
>
> ## LNM and ANCOM will return only OTU count data
> # data.lnm <- LNM.Sim(p, seed = 2, N = 20, dif.otu)
> data.ancom <- ANCOM.Sim(p, seed = 2, N = 20, dif.otu)
> # dim(data.lnm)
> dim(data.ancom)

[1] 40 10

> data.allancom <- cbind(data.ancom %*% taxa.index[, -1], data.ancom)
> dim(data.allancom)

[1] 40 18
```

You also can skipped this step to use the example data we provided as `load(Sim.data)` or package '`MiSPU`' gives.

```
> data("Sim.data")
> summary(Sim.data)

            Length Class      Mode
BIT         100    data.frame list
DTM         100    data.frame list
LNM         100    data.frame list
```

```
ANCOM       100    data.frame list
tree          6    phylo      list
dif.otu      24    -none-     numeric
dif.taxa      5    -none-     numeric
prob.branch   2    data.frame list
group       100    -none-     numeric
```

2.2 Log Ratio Tree Test

Based on the tree, we use the least log ratio transform to do differential analysis. Once the parents is has different probability to his children, which means their children's probability is different, so the ratio must be different between of different group. So, we just do log ratio test of each brothers to decides whether it is differential or not.

```
> data.bit <- as.matrix(Sim.data$BIT)
> grouplabel <- Sim.data$group
> dim(data.bit)

[1] 100 100

> tree <- Sim.data$tree
> p <- min(tree$edge[, 1]) -1
> p

[1] 100

> taxa.index <- Taxa.index(p, tree)
> colnames(data.bit) <- as.character(1:p)
> all.bit <- cbind(data.bit %*% taxa.index , data.bit)
> tree.results <- Tree.ratio(p, tree, taxa.index, all.tab = all.bit ,
+                            group = grouplabel)
> str(tree.results)

List of 4
 $ taxa.pvalue: num [1:99] 0.1843 0.0933 0.5605 0.2286 0.4985 ...
 $ otu.dif    : Named logi [1:100] FALSE FALSE FALSE FALSE FALSE FALSE ...
  ..- attr(*, "names")= chr [1:100] "1" "2" "3" "4" ...
 $ alltab     : num [1:100, 1:199] 165931 64850 28016 187687 66052 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:199] "101" "102" "103" "104" ...
 $ taxa.dif   : chr [1:4] "122" "128" "137" "154"
```

The results of `Tree.ratio` include data after pruned '`alltab`', internal node's pvalue '`taxa.pvalue`', differential internal nodes '`taxa.dif`' which after adjustment on each multiple test, '`dif.otu`' which show is differential one. However, we have to noticed that one child OTU on the leaves of the tree is differential one, it must because his ancestor have different probability for his children, while once ancenstor have differential probability for his children, his children will not be differential one. So here, we have to check the differential OTU finded `Tree.ratio`.

```
> tree.dectected <- Tree.ratio.back(p, tree.ratio = tree.results,
+                       taxa.index, otutab = data.bit, group = grouplabel)
> tree.dectected

 [1] "18" "25" "26" "27" "35" "36" "37" "38" "39" "53" "54" "55" "56" "57"
```

which return the differential OTU by correction step. Some times you may interested the differential internal nodes which can see `result$taxa.dif`.

References

Chao ZHOU, Tao WANG. Differential Abundance Analysis for Microbiome data Incorporating phylogeny. 2018 (Under review).

Mandal, Siddhartha, et al. "Analysis of composition of microbiomes: a novel method for studying microbial composition." Microbial ecology in health and disease 26.1 (2015): 27663.

Xia, Fan, et al. "A logistic normal multinomial regression model for microbiome compositional data analysis." Biometrics 69.4 (2013): 1053-1063.

Torben Tvedebrink (2010). Overdispersion in allelic counts and theta-correction in forensic genetics. Theoretical Population Biology, 78(3), 200-210.