Analysis of Composition of Microbiomes (ANCOM).

Date: June 17, 2015 | Version: 1.1-1

Table of Contents

1. Acknowledgements and References

Authors of the software:

- **R code by:** Dr. Siddhartha Mandal, Norwegian Institute of Public Health, Oslo, Norway.
- **User friendly “shiny” interface developed by:** Dr. Casey Jelsema, Research Fellow, Biostatistics and Computational Biology Branch, NIEHS (NIH), RTP, NC 27709, USA.

References

- Mandal et al. (2015). Analysis of composition of microbiomes: a novel method for studying microbial composition. **Microbial Ecology in Health and Disease** [S.l.], v. 26, ISSN 1651-2235. doi:http://dx.doi.org/10.3402/mehd.v26.27663.

2. Using the shiny application

Data Input

- **Acceptable file types for the data:**

    a. Comma-delimited (.csv),
    b. Tab-delimited (.txt).

- **Two input formats for data:**

    a. Wide format: Subjects on rows, OTUs on columns:

    i. Row 1 should contain the column names (OTU names).
    ii. Last column should contain the group designations (future versions may enable group column to be located elsewhere).
    iii. The rest of the columns should be the OTU / Taxa.

    Example:

OTU1	OTU2	OTU3	Group
###	###	###	###
###	###	###	###

    Where the "###" are abundance counts and group levels.

    b. Tall format: OTUs on rows, subjects on columns:

    i. Column 1 should contain the OTU names.
    ii. Row 1 should contain the group designations.
    iii. The rest of the rows should be the OTU / Taxa.

    Example:

Group	###	###	###
OTU1	###	###	###
OTU2	###	###	###
OTU3	###	###	###

Install packages from R (Use version R 3.2.0 or later versions)

- install.packages("coin")
- install.packages("doParallel")
- install.packages("DT")
- install.packages("foreach")
- install.packages("ggplot2")
- install.packages("Rcpp")
- install.packages("shiny")

To install the above packages follow these instructions:

- In the standard RGui, click *Packages* > *Install package(s) from local zip files …* and then navigate to the source file.
- In RStudio, select *Packages* > *Install Packages*, then set the menu option to "Install From:" to *Package Archive File (.zip; .tar.gz)*, and then navigate to the source file.

Install the ancom.R package:

- Source files:
    a. Windows users: install is the .zip file.
    b. For Linux/Mac users: install .tar.gz file.

Load the ancom.R package:

- library("ancom.R")

Run the shiny application:

- shiny_ancom()

On the shiny_ancom window:

- **Data output:** Two output files are created. The file containing the list of differentially abundant taxa will have a prefix "OTU_list_". The file second file with prefix "Selected_OTU_data_" contains the raw data of corresponding to the taxa that were found to be differentially abundant.
- **Correction for multiple testing:** If "Correct for multiple testing" box is checked then ANCOM performs tests controlling the false discovery rate at

the desired nominal level (specified in the box “Enter significance level (between 0 and 1)” otherwise it does not correct for multiple testing.

- **View side by side boxplots**: To view the side by side boxplots of differentially abundant taxa, click on “Update Plot” once ANCOM finished running.
- **Adjust figure:** Check on this box and choose the desired configurations to view the boxplots of differentially abundant taxa. You may need to use this feature to get a proper resolution of the boxplots.

3. Running ANCOM manually

The ANCOM method can be run manually in R (or other programs which can execute R code) using the function `ANCOM()`. In the R environment, run `?ANCOM` to see the documentation. Some further notes are provided below.

- **Parameters:**
  - `real.data`: the input dataset.
  - `sig`: the significance level (or FDR) at which to run ANCOM.
  - `multcorr`: the type of multiple testing adjustment to be made. There are three options: 1 (a stringent correction), 2 (a less stringent correction), and 3 (no correction). Default behavior is to make no correction (`multcorr=3`). Since option 1 is very stringent, the shiny application uses option 2 for multiple testing corrections.
  - `tau`: a tuning parameter in the Stepwise testing method.
  - `theta`: a tuning parameter in the Stepwise testing method.
- **Tuning parameters (`tau` and `theta`):** The shiny application uses the default values tested in Mandal et al. (2015). For consistency, users are recommended to leave these tuning parameters at their default values, unless they want to explore the performance of ANCOM for different values of the tuning parameters.
- **Data input:** The input data (`real.data`) should match the “wide” format described above. Specifically, subjects should be on the rows, and the columns should be the OTUs (with abundances), and the final column should be the grouping factor.
- **Output:** The output of ANCOM() is a list containing three elements:

- `W`: the values of the test statistic of each OTU. These are used for selecting differentially abundant OTUs and to generate plots. The actual value of `W` may not be relevant for the user.
- `detected`: the names of the differentially abundant OTUs (or a message stating no significant OTUs were detected).
- `dframe`: The input data (to facilitate `plot_ancom()`).

- **Boxplot of differentially abundant OTUs:** `plot_ancom(object)`,where `object` is the output from ANCOM()

4. A small example

The following code will simulate a small dataset and run ANCOM. This is intended to allow the user to see an example of how the input dataset should be structured.

```
nn <- 10
pp <- 20
sim_otu <- matrix( 0, nrow=nn, ncol=pp+1 )
sim_otu <- data.frame(sim_otu)
colnames(sim_otu) <- c( paste0("OTU_", letters[1:pp] ), "Group" )
sim_otu[,pp+1]    <- c( rep("Control",nn/2), rep("Treatment",nn/2)  )
idx_trt <- sim_otu$Group=="Treatment"

for( ii in 1:pp ){
  sim_otu[,ii] <- rpois( nn, 1 )
}

# Create some significance
sim_otu[idx_trt,3] <- rpois( nn/2, 8)
sim_otu[idx_trt,7] <- rpois( nn/2, 8)
sim_otu[idx_trt,9] <- rpois( nn/2, 8)

ancom.out <- ANCOM( real.data = sim_otu, sig = 0.20, multcorr = 2 )
ancom.out$W
ancom.out$detected
plot_ancom(ancom.out)
```