Analysis of Composition of Microbiomes (ANCOM).

Date:	July 14, 2015	Version:	1.1-3

Table of Contents

1. Acknowledgements and References and Notes about ANCOM

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.

Notes on ANCOM:

- ANCOM is based on nonparametric tests, namely, the Kruskal-Wallis and the Friedman tests.
- Applicable to data obtained from standard independent samples design as well as data obtained from a complete block design, such as a repeated measurement design (or longitudinal data).
- For independent samples, ANCOM is based on Kruskal-Wallis test and for data from a complete block design such as the repeated measurement design (or longitudinal data) it is based on the Friedman's test.
- ANCOM does not impute missing values. Hence, samples that are missing at one or more time points in a longitudinal data are omitted from the analysis. ANCOM prints out the ids of missing data that were not used in the analysis.
- ANCOM adds 1 to all OTUs before taking logarithms to deal with OTUs that have zero counts.

2. Using the shiny application

Data Input

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

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

Data Formats:

I. Two input formats for independent samples

c. 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.
iv. Each row represents data corresponding to a sample
v. You don't specify the sample id number but only specify the group the sample belongs to.

Example:

OTU1	OTU2	OTU3	Group
###	###	###	Group1
###	###	###	Group1
###	###	###	Group2
###	###	###	Group2

Where the "###" are abundance counts.

d. 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.
iv. Each column represents data corresponding to a sample
v. You don't specify the sample id number but only specify the group the sample belongs to.

Example:

Group	Group1	Group1	Group2	Group2
OTU1	###	###	###	###
OTU2	###	###	###	###
OTU3	###	###	###	###

II. Two input formats for complete block designs such as repeated measures (longitudinal data)

a. Wide format: Subjects on rows, OTUs on columns. This format is similar to the independent samples except that an additional column of sample ID (or block) needs to be provided as given in the following example.

Example:

OTU1	OTU2	OTU3	Group	Sample ID (or Block)
###	###	###	Group1	ID1
###	###	###	Group2	ID1
###	###	###	Group1	ID2
###	###	###	Group2	ID2

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

b. Tall format: OTUs on rows, subjects on columns. This format is similar to the independent samples except that an additional row of sample ID (or block) needs to be provided.

Example: Example:

Group	Group1	Group2	Group1	Group2
Sample ID (or Block)	ID1	ID1	ID2	ID2
OTU1	###	###	###	###
OTU2	###	###	###	###
OTU3	###	###	###	###

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

```
install.packages("doParallel")
install.packages("DT")
install.packages("exactRankTests")
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. 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. In this software we implement FDR correction as described in Remark 3 of the Supplementary text of Mandal et al. (2015).

- **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, see Remark 3 of the Supplementary text of Mandal et al. 2015), 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.
    - repeated: logical determining whether the data have repeated measures (e.g., longitudinal design).
- **Tuning parameters (**tau and theta**):** For consistency, users are recommended to leave these tuning parameters at their default values provided in the program, 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.
- **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()).

- repeated: the input value for the repeated argument (needed for other functionality).
- n_summary: string summarizing the number of subjects retained / removed in the analysis (for repeated measurement designs).
- sub_drop: string with subject IDs dropped from analysis (for repeated measurement designs).
- sub_keep: string with subject IDs retained in the analysis (for repeated measurement designs).

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

An example

The following code will simulate a small dataset and run ANCOM for independent samples with two groups. 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$Group=="Treatment"
for( ii in 1:pp ){
  sim_otu[,ii] <- rpois( nn, 1 )
}
# Create some significance
sim_otu[idx_trt,3] <- rpois( nn/2, 8)
sim_otu[idx_trt,7] <- rpois( nn/2, 8)
sim_otu[idx_trt,9] <- rpois( nn/2, 8)
ancom.out <- ANCOM( real.data = sim_otu, sig = 0.20, multcorr = 2, repeated=FALSE )
ancom.out$W
ancom.out$detected
plot_ancom(ancom.out)
```