16S Metagenomic Analysis Tutorial

Dr Jun Wang, Rega/VIB
jun.wang@med.kuleuven.be

Introduction:

In this tutorial we will explore the primary characterization of a specific human gut microbiome dataset, namely the enterotype dataset from Arumugam et al (2011). Human microbiome was long considered to be composing of continuous gradient without generalized grouping; yet the enterotype type concept, that microbiomes can be classified into distinct clusters with compositional/functional separations, provided an interesting framework for microbiome studies and their roles in human physiology/diseases. Here we will use the original dataset and use mainly "Vegan" package in R to gather basic descriptions of microbiome data with 16S information and specifically regarding the enterotypes.

Excercise 1: Getting use to R environment

R is a free software environment (that is, can be used on most of the platforms) designed for statistical computing and graphics, it's a language as well as resources, as while main team develops the core (http://www.r-project.org/), scientists all over the world contribute their packages to a specific field (package is a bundle of functions). R can be used directly from command line similar to Python, or can be under Graphic User Interfaces (GUI) like in R studio. Make sure you have R on your computer by typing "R" in command line; or click RStudio to open an R environment.
To start with, check where you are with R (all R commands in this tutorial will start with ">" as appeared on command line, but you only need to type the part after ">"; # will start a commentary that is not part of command but won't influence command either)

```
>getwd()
[1] "/Users/wang"
```

This shows I am at basic folder; if you want to work in a specific folder (where data are/where you want to save things), you can set the working directory:

```
>setwd("/Users/wang/Dropbox/WORK/Practical/")
## this way when you save.image(), the workplace is saved in the folder and will be reloaded next time; or use q(save="yes")
```

Check getwd again and you will find your directory has changed. Now make sure you have the required library by typing:

```
>library(vegan)
Error in library(vegan) : there is no package called ‘vegan’
```

An error message indicates you don't have it yet; you can install the package from CRAN (where most of packages are deposited) by typing:

```
>install.packages("vegan", dependencies=T)
#Dependencies means some other packages Vegan might need
```

After a while vegan should be installed on your computer, and you can load it by library function.

Excercise 2: Data input/output

R works mainly with matrix (tables), and reading/writing table is the most basic functions. We will first load the 16S data from EMBL enterotype website (btw, this is also a nice tutorial with a lot of useful functions and in detail description of enterotyping analysis):
http://enterotype.embl.de/MetaHIT_SangerSamples.genus.txtto your own working folder and read-in data as variable "genus":

```
>genus=read.table("MetaHIT_SangerSamples.genus.txt", header=T,
row.names=1, sep="\t")
## header/row.names define column/row names, while sep="\t" means the
fields are separated by tab; R usually can figure out this by itself, unless there is
mixture of space and tab in one file
```

Or directly from the url:

```
>genus=read.table("http://enterotype.embl.de/MetaHIT_SangerSamples.genus.t
xt", header=T, row.names=1, sep="\t")
```

You can check the genus matrix:

```
>head(genus)
## head checks first few lines
>dim(genus)
## dimesions of genus (i.e. how many rows/columns)
>attributes(genus)
## it will show at $class field that genus is now a data.frame (default by R), let's
say it's similar to matrix but with more structure attached to it
>summary(genus)
## summarize basic stats for genus (per column)
```

As you can figure out, now all samples are organized by column while their genus composition are organized by row. It will lead to trouble in next steps as most of vegan functions need samples organized by row, you can changed this by transposition:

```
>genus=t(genus)
```

Have a look at genus again and you can see it's now organizing samples by row. To output a data frame you can do this:

```
>write.table(genus, "Genus_by_row.tsv", quote=F, sep="\t")
## quote means quotation marks in the table, we don't want that
```

Now in your folder you can see a new file we just saved.

Note: We are not covering the processing of 16S data in this tutorial, as there are various software/workflow devoted to this, and the emergences of new sequencing technologies always demand new analytical approaches. Most influential software so far are Mothur (Schloss et al 2009 Applied Environmental Microbiology) and QIIME (Caporaso et al 2010 Nature methods). We only use already processed, normalized sequences from the enterotye paper.

Exercise 3: Basic stats

We can examine the basic characteristics of sequenced human microbiome by looking at the distribution of all genera and figure out the major members of the communities. Since we have relative abundances (that is, in all samples, all genera add up to 1), we can sort out most abundant genera by first calculate per column mean values:

```
>colMeans(genus)
## similarly you can do rowMeans
```

You can look at the distribution by plotting a histogram:

```
>hist(colMeans(genus))
```

Or sort from lowest to highest by:

```
>sort(colMeans(genus))
## note: genus identifier "-1", corresponds to the fraction of metagenomic reads that cannot reliably be assigned to a known genus, and should not be counted as a genus of itself
```

Now you can pick the most abundant five (or more if you like) genera by binding them to a new data frame:

```
>genera_major=cbind(genus$Bacteroides, genus$Faecalibacterium, genus$Prevotella, genus$Bifidobacterium, genus$Lachnospiraceae)
## you might need to define genus as data.frame again by genus=data.frame(genus), this way you can extract a certain column by its name; when genus is a matrix (check this by attribute function) you can only extract columns by its  number, for instance genus[,1] means first column (while genus[1,] means first row
```

And you can check how much they make up each sample with respect to percentage by:

>rowSums(genera_major)
similarly you can do colSums. As you can see, 5 major genera already represent large part of the community

Now you can have a better visualization by plotting the major genera by:

>barplot(t(genera_major), col=rainbow(5))
barplot takes transposed matrix to plot bars. You can give each section a different color by assigning rainbow(5) -- 5 colors from rainbow instead of default grey scales. You can further refine the barplot in ascending order of one genus with slightly more complex functions
##>barplot(t(genera_major[order(genera_major[,1]),]), col=rainbow(5)), where the samples are ordered (order function) according to value of the first column (Bacteroides)

Excercise 4: Alpha-diversities

Alpha-diversity is one of the essential concepts in ecology, whereas you can describe the richness of communities (how many species you can already find), diversity (how many species are truly there--can be higher than observed numbers especially when it comes to microbial ecology) and evenness (how even are each species relatively to each other), a good summary can be find at http://cran.r-project.org/web/packages/vegan/vignettes/diversity-vegan.pdf , alongside tremendous amount of books in ecology. Today we are only going to cover three major indices used in most of pulications: number of genera (or species), Chao1 index and Shannon evenness.

You can obtain those measures by:
>num_genera=specnumber(genus)
>chao1=estimateR(genus*100000000)[2,]
Chao1 index can be only calculated on integer counts, as the theorum behind it depends on the species only appearing once in one sample. In a lot of cases we actually generate an unified number of counts for each sample (adding up to 1000, 5000 or 10000) to facilitate the calculation. In case of only relative abundances available, we can transform all counts to integers
estimateR generates 5 indices including chao1 (coming out as second row in the result matrix), you can read into this further if interested
>shannon=diversity(genus, "shannon")
diversity function can also calculated several other alpha diversity indices.

Now we are including the enterotypes in to the comparison. As mentioned above, the EMBL enterotyping website provides nice explanations of how enterotypes are calculated and we would not go into details; I will simply provide the results of enterotyping for next round of analysis:

>enterotypes=c("E3_Ruminococceae", "E3_Ruminococceae","E2_Prevotella",
"E3_Ruminococceae", "E3_Ruminococceae", "E3_Ruminococceae",
"E1_Bacteroides", "E2_Prevotella", "E2_Prevotella", "E3_Ruminococceae",
"E3_Ruminococceae", "E3_Ruminococceae", "E1_Bacteroides",
"E3_Ruminococceae", "E3_Ruminococceae", "E3_Ruminococceae",
"E3_Ruminococceae", "E3_Ruminococceae", "E3_Ruminococceae",
"E3_Ruminococceae", "E3_Ruminococceae", "E2_Prevotella",
"E3_Ruminococceae", "E3_Ruminococceae", "E1_Bacteroides",
"E3_Ruminococceae", "E3_Ruminococceae", "E1_Bacteroides",
"E3_Ruminococceae", "E1_Bacteroides", "E1_Bacteroides", "E1_Bacteroides",
"E1_Bacteroides")
this provides a categorical variable (a "factor") for the following anlaysis, "c" function is to provide a list. you can extract a subset using [] as well, for instance enterotypes[1] gives first element while enterotypes[c(1:5,8)] gives first to fifth and 8th elements.

Now you can plot the alpha-diversity measure we just calculated:

>par(mfrow=c(1,3))
this creates a graph of 1 row and 3 panels on the row, instead of plotting new one everytime
>boxplot(num_genera ~ enterotypes, col=rainbow(3), main="Number of genera")
main means the main title
>boxplot(chao1~ enterotypes, col=rainbow(3), main="Chao1 index")
>boxplot(shannon~ enterotypes, col=rainbow(3), main="Shannon evenness")

We want to know if the differences we see in number of genera/chao1 is significant or not, for which we can use anova (analysis of variability) :

>summary(aov(chao1 ~ enterotypes))
summary generate a meaningful report of the result of aov (anova) and as you can see it's significantly different among the three groups

But this is among group comparison (plus anova requires normal distribution of values, which is not always the case with small sample set), for group-wise comparison you might need wilcoxon test (pair-wise, non-parametric--meaning no assumption of normal distributions), the tricky part is to keep two groups only:

> wilcox.test(chao1[c(which(enterotypes=="E1_Bacteroides"),
which(enterotypes=="E3_Ruminococceae"))] ~
enterotypes[c(which(enterotypes=="E1_Bacteroides"),
which(enterotypes=="E3_Ruminococceae"))])
"which" provides a subset of numbers (note it's "=="), and this way we extract chao1 diversity measures of E1 and E3 as well as factors of E1 and E3 (as wilcoxon test requires two levels in the factor)

Alternatively you can just remove one group by putting a "-" sign:

>wilcox.test(chao1[-which(enterotypes=="E2_Prevotella")]~ enterotypes[-which(enterotypes=="E2_Prevotella")])
role of "-" is to remove a subset of numbers from list, leaving the other two groups in the comparison.

Excercise 5: Calculating dissimilarities and perform clustering

As the general trend of "Omics" and "Big data" carries on, increase in quantity of biological observations also lead to more complex, higher-dimensionality of data we have to process. Multi-variate analysis are developed for this kind of task and though firstly in macro-ecology, their principles are applicable for most of the multi-omics studies. To represent the relationships of samples, complex observations are usually simplified to similarity or dissimilarity (distance) between samples. Widely applied distance measures include Euclidean (geometrical distances) , Mahattan (economics), and in ecology particularly Bray-Curtis distance (usually called quantitative measure) and Jaccard (usually called qualitative measure, as only presence/absence is taken into account). A nice summary can be found in section 2.2 in http://cc.oulu.fi/~jarioksa/opetus/metodi/vegantutor.pdf ; and I also recommend to follow this tutorial for detailed tasks in ecological analysis.

We are going to first calculate two distance matrices:
>bray=vegdist(genus, "bray")
>jaccard=vegdst(genus, "jaccard")

Now bray and jaccard are in distance format, which means it's like a matrix but only has values in the lower triangle of the matrix. Usually they have nice correlation to each other when high-depth sampling was done:

>mantel(bray, jaccard)
mantel tests the correlation between distances via permutation (another big field in multi-variate statistics), it shows that the two are highly correlated and this is not by chance (p=0.001)

With distance calculated, one can have a first grasp of how samples look like each other via plotting a dendrogram (clustering), note that several different clustering methods are available:

we still have 3 panels available, otherwise redo par(mfrow=c(1,3))
>plot(hclust(bray, "average"))
>plot(hclust(bray, "complete"))
>plot(hclust(bray, "single"))
hclust means hierarchical clustering, while "average", "complete", "single" are linkage methods (how to define initial clades based on the distances), they differ in concept, practice and results, for a detailed discussion see section 6.1 of vegan tutorial and other discussions on this special topic.

As you can see that the results differ quite a bit, and the consequent interpretation of who is close to whom might be influenced (not to the most closely related samples but more the intermediately related samples). The method of choice and interpretation should be paid special attention, nonetheless clustering is a useful approach for examining major patterns.

With distance another kind of analysis can be done, namely MANOVA (multiple dimensional anova); the analysis is to determine if one kind of factor significantly influences the distances between samples, i.e. if samples belonging to group A (or B respectively) are more similar (homogeneous), while samples from different groups are more distinct. The significance of this test relies (again) on permutation, that each sample is randomly assigned to provided factors and see how many times you can observe a better separation of the groups, e.g. in 1000 permutations you can find only 2 better grouping (more variation explained) than current ones, then the p shall be 0.003; however if you can find a lot more, then the current grouping may be just by random chance.

```
>adonis(bray  ~ enterotypes)
## adonis="analysis of dissimilarity"
```

As you can see, variations explained by enterotypes are 46.91%, meaning roughly half of the total variations can be explained by this kind of grouping, while the rest happens within those 3 groups and might be due to other factors or stochastic processes.

Excercise 6: Ordinations

We have successfully reduced complexity (dimensionalities) of data by clustering, but we also discussed the pitfalls (clustering basically puts all samples on one dimension -- that is x-axis). In practice reducing complexity to two (or three dimensions) are usually desired, and a lot of additional analysis can be based on such "ordination". There is again vast literature and variety of methods available on this topic, and today we are covering the most common ones only.

Two major classes of ordination exist: composition based and dissimilarity based. Understanding both requires a bit imagination beyond the current 3-d space we live in. In composition based ordination (Principal component analysis, PCA), we plot samples based on abundances of species A on axis 1, species B on axis 2, species C on axis 3, and so on; then there are N samples plotted in a very high dimensional space, you can always find a straight line going through the space created a by all these samples and this would be PC1--the most important PC and explains the most variations among all samples, then you can find another line perpendicular to PC1 (remember in multi dimensions there are a lot of lines) that explains second most variations, and PC3 the third most, and so on till N-1s PC. While in distance based ordination (Principal coordinates analysis, PCoA), similar approach is used, with only difference that the multi dimensional space is constructed to fulfill distances between all samples (our 3-d space is usually not enough as Bray-curtis doesn't follow triangle inequality).

>genus_cca=cca(genus)
cca=canonical compoent analysis, one type of PCA; vegan also include rda (redundancy analysis) which transforms data a little bit
>summary(genus_cca)
##it's a long summary of the properties of this ordination, but important part is the "Proportion Explained" at beginning, for it gives "weight" on each axis (CAs) by the percentage they explain individually.
>plot(genus_cca, display="sites")
plot the ordination and only show the samples ("sites" in macro-ecology)
>plot(genus_cca, display="sites", type="p")
show points instead of sample names, don't want the graph to be over-crowded)
>ordispider(genus_cca, enterotypes, lty=2, col="grey", label=T)
draw a connected net of samples belonging to same enterotype. lty defines line type and we use dash line (2) instead of solid lines in order not to over-crowd the graph; use grey lines and show which enterotype they are
also see ordihull, ordiellipse

>genus_cap=capscale(bray~ -1)
this is the vegan version of PCoA, the formular to -1 means "no constrained, free" ordination, more details about constrained are in the easter egg section.
>summary(genus_cap)
>plot(genus_cap, display="sites", type="p")
>ordispider(genus_cap, enterotypes, lty=2, col="grey", label=T)

if you want to change the individual colors/symbols of points you need to do the following:
plot(genus_cap, display="sites", type="n") ## n==not plotting
points(genus_cap, col=self_defined_color, pch=self_defined_symbol) ## self_defined_color/symbol are usually a vector (a list)

Also, statistics can be done on ordination as well, using envfit function, it gives similar results to that of adonis when a factor is tested, but much more powerful when more complex factors are tested (see vegan tutorial on complex meta-data analysis with envfit).

>envfit(genus_cca ~ enterotypes)
>envfit(genus_cap ~ enterotypes)
R2 will be different as ordination is also compression of data, adonis result is more reliable when it comes to total variations but it's limited in power. p-value from envfit is also permutation-based.

Extra

For those who still have some spare time:

Easter Egg: 1: **Try constrained ordination**. Constrained ordination are "hypothesis driven" ordination that put emphasize on some of the factors we want to test.

```
>genus_cap_constrained=capscale(bray ~ enterotypes)
>summary(genus_cap_constrained)
## now you can see first two axes are named "CAP1" "CAP2" and then followed by original MDS. In reality, CAP1 is usually partial of original MDS1 (so does CAP2, you can have multiple CAP based on the factor levels you have), but it is extracted just to show the most important separation by the factors, as a result, MDS1 now has less variation explained
>plot(genus_cap_constrained, type="n")
>points(genus_cap_constrained, col=as.numeric(as.factor(enterotypes)), pch=as.numeric(as.factor(enterotypes)))
>ordispider(genus_cap_constrained, enterotypes, lty=2, col="grey", label=T)
>ordiellipse(genus_cap_constrained, enterotypes, lty=2, col="grey", label=F)
```

Constrained PCA (cca, rda) is not so particular, except that you can do stats on the terms/axis using anova.cca (actually you can also do this on constrained PCoA), for detailed explanations please refer to vegan tutorial.

```
> genus_cca_constrained=cca(genus ~ enterotypes)
> anova.cca(genus_cca_constrained)
> anova.cca(genus_cca_constrained, by="terms")
> anova.cca(genus_cca_constrained, by="axis")
```

Easter Egg 2: Try to use "simper" to identify major contributors to community differences. Simper (similarity percentage) is a test developed by Clarke 1993 Australian Journal of Ecology, aiming at finding species contributing to similarity within a group and dissimilarity between groups (based on bray-curtis distance). Originally implemented in PRIMER-E pacakge (commercial software for windows), it's available in vegan as paired-test only but still valuable to try:

```
>simper(genus[-which(enterotypes=="E2_Prevotella"), ] , enterotypes[-which(enterotypes=="E2_Prevotella")] )
>summary(simper(genus[-which(enterotypes=="E2_Prevotella"), ] , enterotypes[-which(enterotypes=="E2_Prevotella")] ))

## find different taxa between enterotype 1 and 3)
## you can carry on to compare E1 vs E2, E2 vs E3, and summarize the main taxa of differentiation
## perform anova on those and identify the significant ones, plot them
```