Plant vs Animal Diet Workflow |12.01.2019

Mary Jane Espina, Claire Peichel , Sarah Lichtenberger | Group 2 | BioC 5361

Abstract

The topic that our group decided to work on is on plant vs animal diet. This study focuses on the effect of short-term change in diet on the microbiome. The main objective of this work was to recreate some of the figures in the publication by David, 2014. We decided to create two bar plots comparing two diets similar to Figure 1D in the paper. The thing we did differently in our plots is group it by day and also by subject. We also made a dendogram similar to Supp Figure 6A and 6B. iN our figure, we used 16S Sequencing data to recreate the dendogram instead of RNA-Seq which was originally used in the paper.

Sources

Publication: https://www.nature.com/articles/nature12820

TaxaFile: https://knights-lab.github.io/MLRepo/datasets/david/gg/taxatable.txt

Meta Data: https://knights-lab.github.io/MLRepo/datasets/david/mapping-orig.txt

Introduction

Our group directly imported all the data sets from online so we don't need to set the directory and easily reproduce the pipeline.

Our first code is **animal_plant_diet.R** is compose of four code chunks as follows:

- *Data importing* (chunk 1) -> mainly loading all the libraries needed to run the pipeline and importing the data from online sources.
- *Data processing* (chunk 2)-> these chunk include the cleaning of the reads and merging the main data and the metadata
- *Alpha diversity* (chunk 3) -> calculating the alpha diversity, plotting it, and exporting the plot in pdf using ggsave
- *Dendogram* (chunk 4) -> subsetting the baseline and diet data, calculating Spearman correlation, plotting the dendogram and exporting the plot into pdf using base R.

Installation

Make sure the following packages are installed in your computer. And if the its already been installled, load the following libraries:

```
library(tidyverse) # install.packages("tidyverse")--for data wrangling
library(reshape2) # install.packages("reshape2")--to convert data from wide to long format
library(ape) # install.packages("ape") -- to make hierarchal clustering
library(vegan) # install.packages("vegan") -- for making alpha diversity
library(ggplot2) # install.packages("ggplot2") -- to make plots
library(dendextend) # install.pacakges ("dendoextend") --to make dendogram
library(ggpubr) #install.packages ("ggpubr") -- to put together plots in one page
```

Chunk 1- Data Import

Instead of saving the data into a certain directory, we directly import the data from the github repo. We decided to used the taxafile from Green Genes database instead of RefSeq. We named our data as variable Taxa.

```
# Import data --> Taxable from Green Genes Database
Taxa <- read.delim("https://knights-lab.github.io/MLRepo/datasets/david/gg/taxatable.txt",
                   header = TRUE, row.names = 1)
dim(Taxa) #get the dimension of data
glimpse(Taxa) #glimpse of the data set
```

We also dig into the David 2014 paper and found the metadata that mapped into the actual taxafile. We imported the metadata and named it variable Meta.

```
# Import original metadata from David 2014
Meta <- read.delim("https://raw.githubusercontent.com/knights-lab/
                   MLRepo/master/datasets/david/mapping-orig.txt", header = TRUE)
```

Since there other information in the Meta file, we decided to subset only the data that we needed for easy handling which includes the Sample ID, Day, Type of Diet and SubjectFood. To manipulate the data, we use the library 'tidyverse'.

```
# Subset only relevant information from the metadata
Meta <-select(Meta, X.SampleID, SubjectFood, Diet, Day)
```

Chunk 2- Data Processing

In this code chunk, we look at the sums of each column and it's stat to get an idea of the total reads per sample. By doing this, we decided on the cut-off for our quality control. From the data, only samples with total of more than 20,000 reads were included in the downstream analysis.

```
Sum <- colSums(Taxa) # get the column sums of the Taxa data frame
view(Sum) # view the Sum
summary(Sum) # get summary statistics of the colSums

# Quality control of the total reads
# Only samples with more than 20,000 reads were included
TaxaQC <- Taxa[colSums(Taxa)>= 20045]
```

After the QC, we normalized the data by converting it to relative abundance. We used sweep command to replace the data into relative abundance by dividing the value by the total reads. After the data has been normalized, we transpose the data for ease of merging it with the metadata. which is called Merge. The Merge data will be used in downstream analysis.

```
# Convert to relative abundance
TaxaNorm <- sweep(TaxaQC, 2, colSums(TaxaQC), FUN = "/")

# Transpose TaxaNorm data to easily merge two files
TaxaT <- t(TaxaNorm)

row.names(Meta) <- Meta$X.SampleID
Merge <- merge(x=Meta,y=TaxaT,by=0) # merge two data together
```

Chunk 3- Alpha Diversity

Using the previously loaded library 'vegan', we computed alpha shannon diversty. This variable is called TaxaAlpha. In computing the alpga diversity, we used to unmerged file TaxaNorm otherwise the Merge document contains non-numeric vrariables which will give the error.

```
# Compute for shannon-alpha diversity
TaxaAlpha <-  diversity(TaxaNorm,index = "shannon",MARGIN = 2 )
View(TaxaAlpha)
```

The next step will be merging the data alpha diversity file with the meta data. The data contains diet column, which includes the sequencing data of the food. Since this will not be useful in our analysis, we filter out all the information related to food. We also removed redundant columns for clean data. For ease of plotting, we transform our data from wide to long format using melt function. For doing this, we used the 'reshape2' library.

```
# Merge Alpha and Metadata into one file

AlphaMerge <- merge(x=Meta,y=TaxaAlpha,by=0) %>% # merge the alpha diversity and metadata
  filter(!Diet == "NA")  %>% # filter out diet in the data frame
  select(-Row.names) %>% # remove Row.names since it has the same info as X.SampleID
  rename(Subject=SubjectFood) %>% # rename SubjectFood to Subject
  group_by(Day) %>% # Grooup data by day
  melt(id.vars=c("X.SampleID",
                 "Subject","Day", "Diet"),
       variable.name=c("y"), value.name=c("Alpha")) # transform data into long format
```

We plotted the alpha diversity using ggplot. We made a boxlot and faceted the data by diet. In our plot, we included some asthetics like adding the plot title and plot highlights in text.This first plot (AlphaPlot1), we group the data by day summarizing all the subjects alpha diversity per day.

```
AlphaPlot1 <- ggplot(AlphaMerge, aes(x=Day, y=Alpha, group=Day)) +
  geom_boxplot() + # make a boxplot
  facet_wrap (~ Diet) + # facet based on diet
  labs(x="Day", y="Alpha Diversity (Shannon)", # some plot aesthetics and formatting
       title = "Within-species alpha diversity by day for animal vs plant diet",
       subtitle = "There is decline in within-species alpha diversity
       on the 4th day of plant-based diet.") +
  theme_bw() +
  theme(plot.title = element_text(size = 16, margin = margin(b = 10)),
        plot.subtitle = element_text(size = 10, color = "darkslategrey",
                                     margin = margin(b = 25)),
        plot.caption = element_text(size = 8, margin = margin(t = 10), color = "grey70",
                                    hjust = 0))
```

The second plot(AlphaPlot2) has the same format but group the data by subject, summarizing the diversity of all the days per subject.

```
AlphaPlot2 <- ggplot(AlphaMerge, aes(x=reorder(Subject, -Alpha), y=Alpha, fill=Subject)) +
  geom_boxplot() + # make a boxplot
  facet_wrap (~ Diet) + # facet based on diet
  labs(x="Subject", y="Alpha Diversity (Shannon)", # some plot aesthetics and formatting
       title = "Within-species alpha diversity by Subject for animal vs plant diet",
       subtitle = "Each subject has different within-species alpha diversity but pattern
       remains the same in both diet arms.") +
  theme_bw() +
  theme(legend.position = "bottom",
        legend.direction="horizontal",
        plot.title = element_text(size = 16, margin = margin(b = 10)),
        plot.subtitle = element_text(size = 10, color = "darkslategrey",
                                     margin = margin(b = 25)),
        plot.caption = element_text(size = 8, margin = margin(t = 10), color = "grey70",
                                    hjust = 0))
```

After the two plots were created, we wantto save both plots into single pdf file. We used library 'ggpubr' to put together two plots into single page. Since the plot was generated using ggplot2, we use ggsave to save to

file into pdf. The file should be saved in the desktop as alpha.pdf.

```
# put together two plots into a page
alpha <-ggarrange(AlphaPlot1, AlphaPlot2, nrow = 2, ncol = 1) %>%
  ggsave( file="~/Desktop/alpha.pdf", width = 7, height = 10, dpi=300) # save to pdf
```

Chunk 4- Dendogram

We made a heirarchal clustering of 16s rRNA DNA-Seq data. Two plots were made, the baseline diet which we subset the data from Day-4 and the actual diet which we subset the data from Day 4. From the main Merge dat, we filtered only Day -4 data for baseline, and Day 4 for diet data. We also unite the Subject and Diet information. To calculate the Spearman correlation of the matrix, we made the column 'Subject_Diet' into Rownames and transposed the matrix. Same process was followed for both Baseline and Diet dataset.

```
# Subset into Baseline Samples

Baseline <- Merge %>%
  filter(Day == -4) %>% # filter only samples on Day -4
  unite(Subject_Diet, SubjectFood:Diet, sep ="-Subject_") %>% # unite subject and diet column
  select (-Row.names, -X.SampleID, -Day) %>% # remove day and other data
  column_to_rownames(var = "Subject_Diet") %>% # make column into rownames
  t() %>% # transpose the matrix
  cor(method = "spearman") # calculate for spearman correlation


# Subset into Diet Samples
Diet <- Merge %>%
  filter(Day == 4) %>% # filter only samples on Day -4
  unite(Subject_Diet, SubjectFood:Diet, sep ="-Subject_") %>% # unite subject and diet column
  select (-Row.names, -X.SampleID, -Day) %>% # remove day and other data
  column_to_rownames(var = "Subject_Diet") %>% # make column into rownames
  t() %>% # transpose the matrix
  cor(method = "spearman") # calculate for spearman correlation
```

Using 'ape' library, we calculated the disyance matrix and made a hierarchal clustering as well as turn the data into dendogram.

```
# Making heirarchal clustering for baseline and diet

base <- Baseline %>% # baseline data
  dist %>% # calculate a distance matrix,
  hclust(method = "complete") %>% # hierarchical clustering
  as.dendrogram # turn the object into a dendrogram.


diet<- Diet %>%# diet data
  dist %>% # calculate a distance matrix,
  hclust(method = "complete") %>% # hierarchical clustering
  as.dendrogram # turn the object into a dendrogram.
```

To save the plots in this figures, we use base R instead of ggsave since plot were not made using ggplot. Also, graphical parameters were set using par command. We set the number of figures that can fit in a single row by using command mfrow=c(2,1), which means two rows and one column of figures. Also overall margin was set using oma command.

```
#Saving the dendogram  into one pdf file using base R
pdf(file="~/Desktop/phylo.pdf",paper="letter")

# Setting the graphical parameters
par(mfrow =c(2,1), oma=c(0.75,1,0.75,1)) # setting overall margins
```

Parameters of plots were set using mar command. The leaf shape were also set as well as the leaf size and colors. Some aesthetics were also added such as plot and axis labels and font sizes. After the plot was run, the plot was close by dev off function.

```
par(mar=c(4,2.5,1,7)) # plot margins

# Baseline dendogram
base %>% set("leaves_pch", 19) %>% #s et the leaf shape
  set("leaves_cex", 1.2) %>% # set leaf size
  set("leaves_col",  value = c("purple", "purple", "purple", "purple",
                               "salmon","purple", "salmon","purple",
                               "purple", "purple", "purple", "purple",
                               "salmon", "purple")) %>% # manually assigning colors
  set("labels_cex", 0.75) %>% # font size of the label
  plot(main = "16s rRNA DNA-Seq Day -4 (Baseline)", # header of the plot
       xlab= "Spearman distance", horiz=TRUE) # label of x-axis

par(mar=c(4,2.5,2.5,7)) # plot margins

# Diet dendogram
diet %>% set("leaves_pch", 19) %>% # set the leaf shape
  set("leaves_cex", 1.2) %>% # set leaf size
  set("leaves_col",  value = c("salmon","salmon", "salmon", "salmon", "salmon", "salmon",
                               "salmon", "purple", "purple", "purple", "purple", "purple",
                               "purple", "purple")) %>% # manually assigning colors
  set("labels_cex", 0.75) %>% # font size of the label
  plot(main = "16s rRNA DNA-Seq Day 4 (Diet)", # header of the plot
       xlab= "Spearman distance", horiz=TRUE) # label of x-axis

dev.off() # turning device-off
```

After the code above, there will be two plots saved in your desktop. The alpha.pdf for alpha diversity and the phylo.pdf for the dendogram. The codes above was in an R code animal_plant_diet.R

Run everything using source code

To make sure that everything is working and the workflow was automated, download the two R scripts into your desktop.

- First script: **animal_plant_diet.R**
- Second script: **run_analysis.R**

Open a fresh R environment and run the second script by sourcing the first code.

```
source("~/Desktop/animal_plant_diet.R")
```

Code availability

Codes can be accessed in my github account: https://github.com/mjayespina/BioC5361Project