Downstream analysis in R

Sept 28, 2021

This is the R markdown documentation report for the Analysis in R

First load the R packages necessary for the analysis

```
library(taxonomizr)
library(tidyverse)
library(tidyr)
library(readr)
library(seqRFLP)
library(phyloseq)
library(purrr)
library(magrittr) # necessary for exporting data
library(ggplot2)      # graphics
library(readxl)       # necessary to import the data from Excel file
library(dplyr)        # filter and reformat data frames
library(tibble)
library(vegan)
library(HTSSIP)


# read in the data and create phyloseq object

otu <- read.csv("otua.tsv",header = TRUE, sep = "\t")
```

treating the Otu object

removed the first row in the otu object

```
row_removed <- otu %>% slice(-c(1))
```

#FLip the otu dataframe

```
flipped<- data.frame(t(row_removed[-1]))
colnames(flipped) <- row_removed[,1]
```

#Transform into matrixes otu and tax tables

```
otu_mat <- as.matrix(flipped)
```

Taxonomy from qiime

```
taxonomy_table <- read.table(file = 'taxonomy_qiime2.tsv', sep = '\t', header = TRUE)
```

taxonomy table treatment

Filtering unwated features

```
filtered_tax <- taxonomy_table[ !grepl("Mitochondria", taxonomy_table$Taxon) , ]
filtered_tax <- filtered_tax[ !grepl("Unassigned", filtered_tax$Taxon) , ]
```

treatment of the assigned

Renaming the first column

```
separate_DF2 <- filtered_tax %>% separate(Taxon, c("Kingdom", "Phylum", "Class", "Order", "Family", "Ge
```

#remove confidence column

```
separate_DF2 <- subset(separate_DF2,select = -c(Confidence))
separate_DF2 <- separate_DF2[,-9]
```

#removing prefixes that start with a D

```
separate_DF2$Kingdom <- gsub("D_0__","",as.character(separate_DF2$Kingdom))
separate_DF2$Phylum <- gsub("D_1__","",as.character(separate_DF2$Phylum))
separate_DF2$Class <- gsub("D_2__","",as.character(separate_DF2$Class))
separate_DF2$Order <- gsub("D_3__","",as.character(separate_DF2$Order))
separate_DF2$Family <- gsub("D_4__","",as.character(separate_DF2$Family))
separate_DF2$Genus <- gsub("D_5__","",as.character(separate_DF2$Genus))
separate_DF2$Species <- gsub("D_6__","",as.character(separate_DF2$Species))
```

#separating assigned and unassigned # preblast = samples assigned taxonomy by silva in qiime

```
pre_blast1 <- separate_DF2 %>% filter(!is.na(separate_DF2$Species))
assigned1 <- separate_DF2 %>% filter(is.na(separate_DF2$Genus))
```

Reading in the feature sequences

```
feature.sequences <- read.csv("/opt/data/oscarmwaura/transcriptome/bac-16S/round-2OTU-pipeline/merged/f
```

renaming the columns

```
feature.sequences <- feature.sequences %>% rename(Feature.ID=V1, Sequences=V2)
```

merging the unassigned feature ids with the sequences for blasting

```
merged <- merge(feature.sequences, assigned1, by="Feature.ID", all = FALSE)
```

removed the rest of the columns

```
merged <- merged[,c(1,2)]
```

converting the merged data into a fasta file for blasting

```
my_blast_sequences <- dataframe2fas(merged, file = "unassigned.fasta")
```

Running Blast and Taxonomy

#Running blast on the server

```
blastn = "/opt/apps/blast/2.10.1+/bin/blastn"
blast_db = "./blast_dir/16S_ribosomal_RNA"
input = "./unassigned.fasta"
evalue = 9.6e-6
format = 6
max_target = 1
colnames <- c("qseqid",
              "sseqid",
              "evalue",
              "bitscore",
              "sgi",
              "sacc")
blast_out <- system2("/opt/apps/blast/2.10.1+/bin/blastn",
                     args = c("-db", blast_db,
                              "-query", input,
                              "-outfmt", format,
                              "-evalue", evalue,
                              "-max_target_seqs", max_target,
                              "-ungapped"),
                     wait = TRUE,
                     stdout = TRUE) %>%
  as_tibble() %>%
  separate(col = value,
           into = colnames,
           sep = "\t",
           convert = TRUE)
```

getting taxonomyID from the NCBI accessions

```
unassigned <- accessionToTaxa(c(blast_out$sseqid),"accessionTaxa.sql")
```

getting the taxonomic classification from the taxonomy IDs

```
classification <- getTaxonomy(unassigned,'accessionTaxa.sql')
```

appending the feature IDs to the blast output

```
combined <- cbind(blast_out$qseqid, classification)
combined2 <- as.data.frame(combined)
```

renaming the unassigned columns

```
new_table <- combined2 %>% dplyr::rename(Feature.ID = V1, Kingdom = superkingdom, Phylum = phylum, Clas
updated_table <- rbind(new_table, pre_blast1)
```

#checking duplicates in the taxonomic tables

```
duplicated(updated_table)
n_occur <- data.frame(table(updated_table$Feature.ID))
not_duplicate <- updated_table[!duplicated(updated_table), ]
write_tsv(joined,"joinedtaxonomy.tsv")
```

get column Feature and append it to the flipped table

```
Feature.ID <- feature.sequences[1]
appended <- cbind(Feature.ID, flipped)
```

#updating feature_table

```
feature_tax <- merge(appended, not_duplicate, by="Feature.ID", all = FALSE)
#Extracting the feature table to get the updated table
feature_table <- feature_tax[,c(-2:-6)]
```

generating a matrix for the taxonomy and feature tables for creating a phyloseq object

#taxonomy table

```
my_taxonomy <-  feature_table %>% remove_rownames %>% column_to_rownames(var="Feature.ID")
my_taxonomy <- as.matrix(my_taxonomy)
TAX = tax_table(my_taxonomy)

#OTU table / feature table
OTU_table <- feature_tax[,c(1:6)]
my_otu_table <- OTU_table %>% remove_rownames %>% column_to_rownames(var="Feature.ID")
my_OTU_table2 <- data.matrix(my_otu_table)
class(my_OTU_table2)
OTU <- otu_table(my_OTU_table2, taxa_are_rows = TRUE)
```

#Reading the sample meta data into R

```
sample_metadata <- read.csv("../../transcriptome/metadata/BSF-Metadata-File.tsv", sep = '\t', header = T
supplementary <- sample_metadata[!(sample_metadata$X.SampleID=="#q2:types"), ]
sdata <- supplementary %>% remove_rownames %>% column_to_rownames(var="X.SampleID")
samdata <- sample_data(sdata)

#creating a phyloseq object
physeq <- phyloseq(OTU, TAX, samdata)
physeq
```

Organisms present

```
phyla <- get_taxa_unique(physeq, "Phylum")
view(phyla)
genus <- get_taxa_unique(physeq, "Genus")
view(genus)
class <- get_taxa_unique(physeq, "Class")
view(class)
sp_names <- get_taxa_unique(physeq, "Species")
view(sp_names)
```

Generating the Taxa bar plot

removing OTUs boundaries

```
plot_bar(physeq, fill = "Species") +
  geom_bar(aes(color=Species, fill=Species), stat="identity", position="stack")
```

alpha diversity

```
plot_richness(physeq, measures=c("Chao1", "Shannon"))
plot_richness(physeq, measures=c("Chao1", "Shannon"), x="DIET", color="DIET")
```

PCOA

```
physeq.ord <- ordinate(physeq, "NMDS", "bray")
carbom_fraction <- merge_samples(physeq, "DIET")
plot_bar(carbom_fraction, fill = "Phylum") +
  geom_bar(aes(color=Phylum, fill=Phylum), stat="identity", position="stack")
```

beta diversity attempt

beta diversity

```
library("ggpubr")
```

```
library("tidyr")
library("phyloseq")
library("ggplot2")
library("dplyr")
library("ggpubr")
library("vegan")
library(lattice)
library(permute)
ord = ordinate(relab_genera, method="PCoA", distance = "bray")


plot_ordination(relab_genera, ord, color = "DIET", shape="Genus") +
  geom_point(size=4) +
  stat_ellipse(aes(group=DIET))
```