Wk1 Microbiome Analysis Intro

Christine V. Hawkes

1/12/2022

Contents

Overview 1

R Packages 1

Connect RStudio to GitHub 3

Import data into R from the Class GitHub Data Repository 4

Practice file manipulation 6

Coding Exercises 8

Session Info 8

Overview

Today we'll go through some basics:
* work with R packages
* connect to GitHub
* open data files
* manipulate data files

R Packages

Install packages - three different repository types

Install R packages from CRAN

```
install.packages("devtools")
install.packages("Biostrings")
install.packages("seqinr")
install.packages("tidyverse")
install.packages("data.table")
install.packages("magrittr")
install.packages("rmarkdown")
install.packages("knitr")
install.packages("caTools")
install.packages("vegan")
```

Install R packages from Bioconductor

```r
# phyloseq
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("phyloseq")

# dada2
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("dada2")
# if this does not work, see other options at
# https://benjjneb.github.io/dada2/dada-installation.html

# DeSeq2
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("DESeq2")
```

Install packages from GitHub

```r
# Lulu
library(devtools)
install_github("tobiasgf/lulu")
```

Other useful package commands

```r
# Missing dependencies?
# CRAN
# install.packages("additional_missing_package_2")
# Bioconductor
# BiocManager::install("additional_missing_package_1", version = "3.12")
# example:
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("ShortRead")

# See what packages are installed - can also use Rstudio Packages tab
library()

# load a package and determine the software version
library(vegan); packageVersion("vegan")

# find the citation for package
citation("vegan")

# get help with packages
packageDescription("vegan")
help(package="vegan")
ls("package:vegan") # lists package functions
help(anosim, package="vegan") # help with specific function
browseVignettes(package="vegan") # locally or online
```

```r
# delete packages
remove.packages("packagename")
```

Connect RStudio to GitHub

1. From your computer

- Create a single local folder where you will keep all of your course materials

2. From GitHub

- Sign in to github.ncsu.edu with your unity ID
- Navigate to GitHub repository with your last name inside MicrobiomeAnalysis2022 organization
- Edit the ReadMe document to suit your preferences
- Select "clone with HTTPS" from the Code menu and copy the address
- Note: HTTPS requires you to sign in with your unity ID for NCSU authentication
  - If you want to push/pull without sign in, see instructions for using SSH keys instead of HTTPS
  - https://docs.github.ncsu.edu/github-best-practices/

3. From RStudio, clone the GitHub Repo

- Make sure Git is installed on your local computer
  - Problems? Check https://happygitwithr.com/install-git.html
- Start a new project *File > New Project > Version Control > Git*
- In "Repository URL", paste the URL of your new GitHub repository.
- Create Project from existing directory - select the folder on your computer from step 1
- New Git menu pops up at top of RStudio; as you work:
  - use "commit" to save changes to GitHub repo
  - use "push" to send local changes to GitHub repo
  - use "pull" to update your local files if changes were made from another computer

4. MicrobiomeAnalysis2022/ClassDocs and /ClassData

- You have access to both of these repos

- All files can be downloaded using R (see next section)

- If you want a copy, you can clone to your local computer as above

- However, no changes that you make will be allowed back in the main repo

5. Problems? Check

- https://happygitwithr.com/rstudio-git-github.html

- https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN

6. Open a new .Rmd file to work in and save it as "Wk1_YourLastName.Rmd"

- Check the Git window to the top right of R studio and make sure this file has appeared

- Continue below by copying and pasting code into your new file

---

Import data into R from the Class GitHub Data Repository

- to find the location of any GitHub file, click on that file in GitHub, select raw, and copy the URL

- download a csv file from github and assign it to a file in R ("mydata")

- multiple approaches can be used to import files
- today we will use the file **wk1_testdata.csv** to practice

1. open file with baseR::read.csv

```
mydata1 <- read.csv(url("https://raw.github.ncsu.edu/MicrobiomeAnalysis2022/ClassData/main/wk1_testdata
                    stringsAsFactors=FALSE, header=FALSE)
# "<-" assigns the variable mydata1 to the data frame we read in using read.csv;
# use this instead of "=" since it has operator precedence and is only used for assignment
mydata1
```

```
#   V1  V2 V3
# 1  1  50  3
# 2  2 100  3
# 3  3 200  4
```

```
str(mydata1) #returns data structure
```

```
# 'data.frame': 3 obs. of  3 variables:
#  $ V1: int  1 2 3
#  $ V2: int  50 100 200
#  $ V3: int  3 3 4
```

2.open with readr::read_csv

- must specify filetype, but parses file structure

```
library(readr)
mydata2 <- readr::read_csv(url("https://raw.github.ncsu.edu/MicrobiomeAnalysis2022/ClassData/main/wk1_t
                    col_names=FALSE)
mydata2
```

```
# # A tibble: 3 x 3
#      X1    X2    X3
#   <dbl> <dbl> <dbl>
# 1     1    50     3
# 2     2   100     3
# 3     3   200     4
```

```
str(mydata2)
```

```
# spec_tbl_df [3 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
#  $ X1: num [1:3] 1 2 3
#  $ X2: num [1:3] 50 100 200
#  $ X3: num [1:3] 3 3 4
#  - attr(*, "spec")=
#   .. cols(
#   ..   X1 = col_double(),
#   ..   X2 = col_double(),
#   ..   X3 = col_double()
#   .. )
#  - attr(*, "problems")=<externalptr>
```

```
# other common file options with readr
# readr::read_tsv() for tab separated files
# readr::read_delim() for general delimited files
# read_csv and read_tsv are special cases of read_delim
```

3. open with data.table::fread

- much faster for very large files, detects file type and columns/rows

```
library(data.table)
mydata3 <- data.table::fread("https://raw.github.ncsu.edu/MicrobiomeAnalysis2022/ClassData/main/wk1_tes
                 stringsAsFactors=FALSE, header=FALSE)
mydata3
```

```
#    V1  V2 V3
# 1:  1  50  3
# 2:  2 100  3
# 3:  3 200  4
```

```
str(mydata3)
```

```
# Classes 'data.table' and 'data.frame':    3 obs. of  3 variables:
#  $ V1: int  1 2 3
#  $ V2: int  50 100 200
#  $ V3: int  3 3 4
#  - attr(*, ".internal.selfref")=<externalptr>
```

4. Open other data formats

- For .html, .Rmd, .R, etc., download files from GitHub with download.file
- Today, use this command to download the syllabus or similar from GitHub

```
download.file(url = "https://rawgithubfilelocation",
              destfile = ".../yourlocalfolder/filename")
```

5. other useful checks on data files

```
anyNA(mydata3)    # look for "NA" in the data file
colnames(mydata3) # look at column names
rownames(mydata3) # look at row names
rm(mydata3)       # delete the file
```

Practice file manipulation

1. Use methods above to open "wk1__TestASVs.csv" from the ClassData repo

- open the file "testASVs.csv" from GitHub
- check row and column specifications in your code to open it
- check row and column names after opening the file

```
testASVs<- readr::read_csv(url("https://raw.github.ncsu.edu/MicrobiomeAnalysis2022/ClassData/main/wk1_Te
                           col_names=TRUE)
```

2. Summarize the data using dplyr

- open tidyverse package

```
library("tidyverse")
```

3. Sum and view ASVs (columns)

```
#   NOTE: the "%>%" pipe operator "pipes" the value forward into the expressions that follow
  # (i.e., do this, then do this next)
#   an alternative would be to write nested functions - either work

# use dplyr built-in functions to sum ASV columns
# https://dplyr.tidyverse.org/index.html

ASV_sums <- testASVs %>%
  dplyr::summarise_if(is.numeric, # ignores non-number columns
                      sum,        # sum the columns
                      na.rm=TRUE) # skips NA values
ASV_sums
```

```
# # A tibble: 1 x 6
#    ASV1  ASV2  ASV3  ASV4  ASV5  ASV6
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
# 1    22    25    42    14    22     1
```

```
#   note which ASVs have sums <=1
#   can repeat the above to calculate mean and sd for each ASV
```

4. Sum and view samples (rows)

```
Sample_sums <- testASVs %>%
  dplyr::select(ASV1:ASV6) %>%  # select the columns to sum across
  rowSums(na.rm=TRUE)           # sum the rows
Sample_sums                     # note which samples have sum = 0
```

```
# [1] 23 19 32 23 29  0
```

5. Filter the data by name to remove specific singleton ASVs and samples with no ASVs

```
ASV_nosing <- testASVs %>%
  dplyr::filter(Sample != "soil6") %>% # remove the row named "soil6"
  dplyr::select(-ASV6)                 # remove the column named "ASV6"
ASV_nosing
```

```
# # A tibble: 5 x 6
#   Sample  ASV1  ASV2  ASV3  ASV4  ASV5
#   <chr>  <dbl> <dbl> <dbl> <dbl> <dbl>
# 1 soil1     10     0     4     0     9
# 2 soil2     12     0     7     0     0
# 3 soil3      0     6    12    14     0
# 4 soil4      0     8    15     0     0
# 5 soil5      0    11     4     0    13
```

6. Filter using the summed values to remove (useful when you have large files)

```
ASV_nosing2 <- testASVs %>%
  dplyr::mutate(sum_sample = rowSums(select(., starts_with("ASV")))) %>%
  dplyr::filter(sum_sample!=0) %>% ## remove rows w/ sum of zero
  dplyr::select(-sum_sample)  %>%  ## delete row that was added
  dplyr::select_if(negate(function(col) is.numeric(col) && sum(col) <= 1)) #remove cols w/ sum<0
ASV_nosing2
```

```
# # A tibble: 5 x 6
#   Sample  ASV1  ASV2  ASV3  ASV4  ASV5
#   <chr>  <dbl> <dbl> <dbl> <dbl> <dbl>
# 1 soil1     10     0     4     0     9
# 2 soil2     12     0     7     0     0
# 3 soil3      0     6    12    14     0
# 4 soil4      0     8    15     0     0
# 5 soil5      0    11     4     0    13
```

```
## compare ASV_nosing2 to ASV_nosing - are they the same?
```

Coding Exercises

Manipulate files as described below. For additional help, follow the links provided and see cheat sheets in GitHub/MicrobiomeAnalysis2022/ClassDocs. Complete what you don't finish in class for homework. Coding exercises do not need to be uploaded this week - this is for basic skill-building in handling data in R with an emphasis on dplyr. 1. create a new data file with dimensions of 10x10

- (9x9 of data, 1 row for column headings and 1 column for sample names)

2. rename a column (imagine you find a mistake in your file)

- for help see https://dplyr.tidyverse.org/reference/rename.html#examples

3. add a column with at least three categories (e.g., group=A,B,C) using baseR, tibble::add_column, or dplyr::mutate

- https://tibble.tidyverse.org/reference/add_column.html
- https://dplyr.tidyverse.org/reference/mutate.html

4. repeat this on the original file by creating a data frame with the groups and combining data sets

5. summarize your data (sum, mean, sd) overall and by the categories in the new column

6. if you found 1-5 very easy, try to convert from short format to long format data using tidyr::pivot_longer and then back again using tidyr::pivot_wider

- https://tidyr.tidyverse.org/reference/pivot_longer.html
- https://tidyr.tidyverse.org/reference/pivot_wider.html

8. save at least one data file under a new name using the write.csv command and push to GitHub

---

Session Info

```
sessionInfo()
```

```
# R version 4.1.2 (2021-11-01)
# Platform: x86_64-w64-mingw32/x64 (64-bit)
# Running under: Windows 10 x64 (build 19042)
#
# Matrix products: default
#
# locale:
# [1] LC_COLLATE=English_United States.1252
# [2] LC_CTYPE=English_United States.1252
# [3] LC_MONETARY=English_United States.1252
# [4] LC_NUMERIC=C
# [5] LC_TIME=English_United States.1252
#
# attached base packages:
# [1] stats     graphics  grDevices utils     datasets  methods   base
#
# other attached packages:
#  [1] forcats_0.5.1     stringr_1.4.0     dplyr_1.0.7       purrr_0.3.4
#  [5] tidyr_1.1.4       tibble_3.1.6      ggplot2_3.3.5     tidyverse_1.3.1
#  [9] data.table_1.14.2 readr_2.1.1
#
# loaded via a namespace (and not attached):
#  [1] tidyselect_1.1.1 xfun_0.28        haven_2.4.3      colorspace_2.0-2
#  [5] vctrs_0.3.8      generics_0.1.1   htmltools_0.5.2  yaml_2.2.1
#  [9] utf8_1.2.2       rlang_0.4.12     pillar_1.6.4     withr_2.4.3
# [13] glue_1.5.0       DBI_1.1.1        bit64_4.0.5      dbplyr_2.1.1
# [17] readxl_1.3.1     modelr_0.1.8     lifecycle_1.0.1  cellranger_1.1.0
# [21] munsell_0.5.0    gtable_0.3.0     rvest_1.0.2      evaluate_0.14
# [25] knitr_1.36       tzdb_0.2.0       fastmap_1.1.0    parallel_4.1.2
# [29] curl_4.3.2       fansi_0.5.0      Rcpp_1.0.7       broom_0.7.10
# [33] backports_1.4.0  scales_1.1.1     vroom_1.5.7      jsonlite_1.7.2
# [37] fs_1.5.2         bit_4.0.4        hms_1.1.1        digest_0.6.28
# [41] stringi_1.7.6    grid_4.1.2       cli_3.1.0        tools_4.1.2
# [45] magrittr_2.0.1   crayon_1.4.2     pkgconfig_2.0.3  ellipsis_0.3.2
# [49] xml2_1.3.3       reprex_2.0.1     lubridate_1.8.0  assertthat_0.2.1
# [53] rmarkdown_2.11   httr_1.4.2       rstudioapi_0.13  R6_2.5.1
# [57] compiler_4.1.2
```