Installation instructions: 16S and metabolomics analyses

Maureen Carey

R - the language

R is a statistical programming language that has been widely used by biologist and other applied scientists.

Follow the instructions below. If you already use R, feel free to skip ahead to the RStudio - a useful tool or R packages - the vocabulary sections. Please note, however, that some packages may not be available with older R versions. If you encounter these issues, just update your R installation. [1]

Mac Users

Installing R on a Mac:

1. Open an internet browser and go to www.r-project.org.
2. Click the "download R" link under "Getting Started."
3. Select a CRAN location (a mirror site) and click the corresponding link. I always use the University of Kansas mirror out of habit, but you can pick any! Because R is a free resource and so many people use it, mirroring is used to share resources (i.e servers).
4. Click on the "Download R for (Mac) OS X" link at the top of the page.
5. Click on the file containing the latest version of R under "Files." This will be at the top of the "Files" section.
6. Save the .pkg file, double-click it to open, and follow the installation instructions.
7. Move on to the RStudio - a useful tool section.

Windows Users

Installing R on a Windows machine:

1. Open an internet browser and go to www.r-project.org.
2. Click the "download R" link under "Getting Started."
3. Select a CRAN location (a mirror site) and click the corresponding link. I always use the University of Kansas mirror out of habit, but pick any! Because R is a free resource and so many people use it, mirroring is used to share resources (i.e servers).
4. Click on the "Download R for Windows" link at the top of the page.

5. Click on the "install R for the first time" link at the top of the page. This will be at the top of the "Files" section.
6. Click "Download R for Windows" and save the executable file somewhere on your computer. Run the .exe file and follow the installation instructions.
7. Move on to the RStudio - a useful tool section.

[1] Instructions for R - the language and RStudio - a useful tool are adapted from the material provided by the UTAustinX 'Foundations of Data Analysis' course.

RStudio - a useful tool

RStudio makes editing and running R code easier and we recommend you use it. RStudio is an integrated development environment (IDE) for R and enables you to write and execute code. Think of it like Microsoft Word: you don't *have* to write text documents in Word, but Word layers on many extra functions that make it easier to do so.

Mac Users

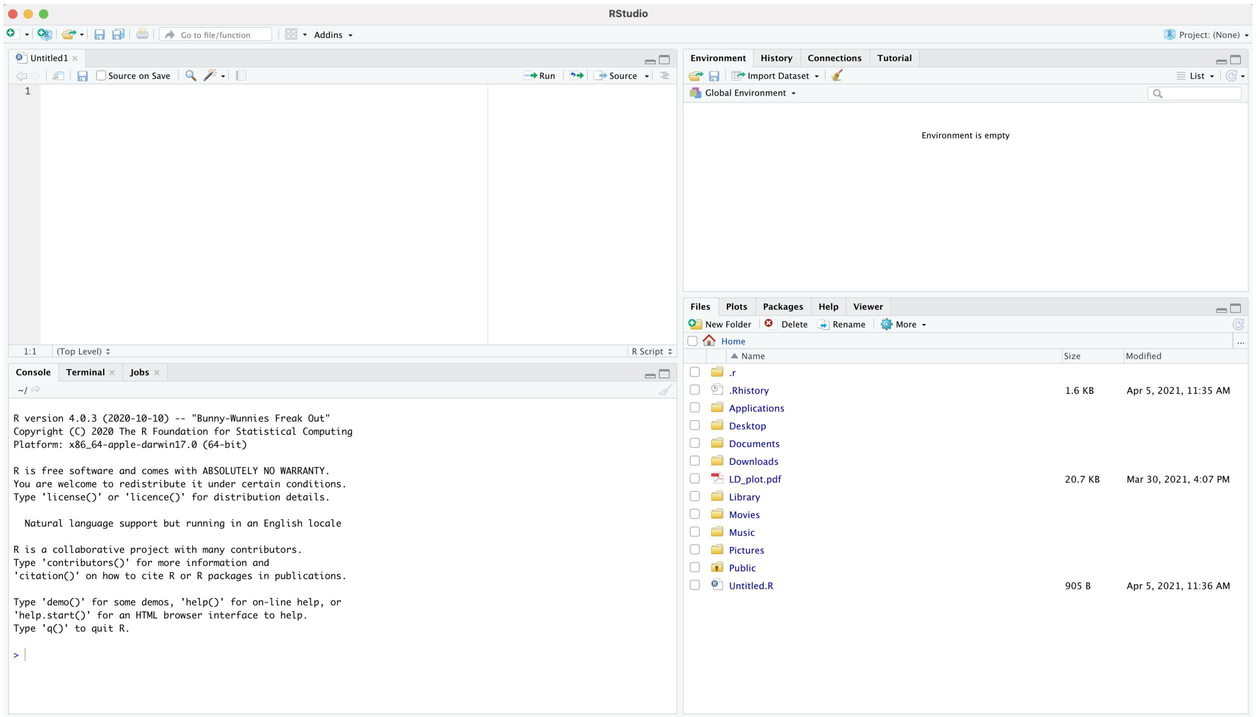
1. Go to www.rstudio.com and click on the "Download" button at the very top of the blue banner.
2. Click on "Download RStudio Desktop" (this is the free version for individual users).
3. Click on the version recommended for your system, or the latest Mac version, save the .dmg file on your computer, double-click it to open, and then drag and drop it to your applications folder.

Windows Users

1. Go to www.rstudio.com and click on the "Download" button at the very top of the blue banner.
2. Click on "Download RStudio Desktop" (this is the free version for individual users).
3. Click on the version recommended for your system, or the latest Windows version, and save the executable file. Run the .exe file and follow the installation instructions.

Using RStudio

When you open RStudio, you will see something like this:



Environment is empty

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Name	Size	Modified
.r		
.Rhistory	1.6 KB	Apr 5, 2021, 11:35 AM
Applications		
Desktop		
Documents		
Downloads		
LD_plot.pdf	20.7 KB	Mar 30, 2021, 4:07 PM
Library		
Movies		
Music		
Pictures		
Public		
Untitled.R	905 B	Apr 5, 2021, 11:36 AM

Figure 1: Screenshot of RStudio window

If you don't see the top left panel, go to File > New File > R Script and then you will see all 4 quadrants of the window. The top left is a R script - where you can save your code into a file. The bottom left is the console. This is where you execute code (after the '>'). The top right panel will show you what variables are loaded into the environment. Try copying and pasting this code into the console and hitting enter:

```
a = 1
```

Variable 'a' will now be in your environment!

The bottom right panel is a workhorse - it may show folders on your computer, but it can also show you plots that you've made and help you access help pages for R and individual packages (more on those later). Try running this to see what happens when you querry the help pages for the term 'mean':

```
?? mean
```

R packages - the vocabulary

If R is a language and RStudio is your 'Word document' to communicate in that language, then R packages are new vocabulary. Yes, you could describe DNA as 'tiny chemicals that tell cells what they can do', but it's much more efficient to call it 'DNA.'

R packages are developed by scientists and software developers to extend base R (the built-in vocabulary and functions) to more efficiently do additional tasks.

Packages for the TUMI course

This week we will use the following packages (in no particular order):

- Packages for visualization/data wrangling:
  - tidyverse
  - ggpubr
- Packages for sequence analysis:
  - DADA2
  - phyloseq
  - Biostrings
- Packages for interpretive analyses:
  - mixOmics
  - randomForest
  - ape
  - vegan
  - pROC
- Other packages:
  - readxl
  - tictoc

The end of this document provides step-by-step directions.

How do I find and install R packages?

Well-established packages can be installed from one of two places, depending on where it is shared:

1. CRAN
2. Bioconductor

(Packages that are actively being developed may also be available via GitHub but we will not be using any of these this week.)

Bioconductor packages tend to be more specific to biological applications (think: sequence analysis) whereas CRAN packages tend to be more field-independent (think: statistics or visualization).

To install a Bioconductor package, you would use this format:

```
source("https://bioconductor.org/biocLite.R")
biocLite("dada2")
```

To install a package from CRAN (which stands for the "Comprehensive R Archive Network"), you would use this format:

```
install.packages("ggplot2")
```

Fortunately, there is an approach that let's you install any of these packages without having to know where they came from:

```
BiocManager::install(c("dada2","ggplot2"))
```

To look at what packages you have installed (and which have been pre-installed as part of you R download), do this:

```
installed.packages()
```

(I have a LOT - pages and pages - so I am not going to show my results.)

Packages only need to be installed before you use them for the first time. We've asked you to try these installations before the class starts! (Remember we will be available for installation support only on 04/09/2021.) Every time you use a package, however, you will need to 'load' it in for active use. Here is how you load a package:

```
library("ggplot2")
```

Generally, I recommend you install one package at a time, like this:

```
install.packages("ggplot2")
```

Installing one at a time lets you easily identify if one package's installation has any errors, whereas this is harder to identify if all are run at once. That being said, you can install many at once like this:

```
install.packages(c("ggplot2","tidyverse"))
```

TL;DR Installation instructions for the TUMI course

Ok now let's actually install them! Copy and paste this into your RStudio console:

```
install.packages("BiocManager")
BiocManager::install(version = '3.12')
BiocManager::install(c("dada2","phyloseq","Biostrings","mixOmics",
                       "tidyverse","randomForest","ape","tictoc",
                       "pROC","readxl","ggpubr","vegan"))
```

You may be asked, "Update all/some/none? [a/s/n]:" Please enter "a" to indicate that R can install/update any necessary packages used by the packages you are actively installing.

Additionally, you may be asked, "Do you want to install from sources the package which needs compilation? (Yes/no/cancel)". You can again enable this by typing "yes".

FYI the "tidyverse" packages is actually a collection of packages, including ggplot2, dplyr, tidyr, readr, purrr, tibble, stringr, and forcats.

Keep any eye out for error messages. If you get any, ask for installation help and provide us with the following info:

1. R version
2. what did you try?

3. what error message did you get?

I recommend testing your installations by loading each package BEFORE the course begins. Again, to load a package, you enter the following line in the console of your RStudio window for each package:

```r
library("package_name")
```

One last comment

While I've got your attention, I think it's important to discuss how to cite packages! You wouldn't use a published protocol without citing it, so do the same with your computational work. To find the reference for an individual package, try this:

```r
citation("mixOmics")
```

You can also search R packages on Google Scholar.

See you next week!