R community analysis

Matthew R. Gemmell

2024-08-09

Contents

Intro

Chapter 1

Introduction

A lot of different analyses and visualisations can be carried out with community data. This includes taxonomy and functional abundance tables from 16S rRNA and Shotgun metagenomics analysis. This workshop will teach you how to use R with the `phyloseq` R object; a specialised object containing an abundance, taxonomy, and metadata table.

The workshop will use a 16S dataset that has been pre-analysed with QIIME2 to create the ASV table, taxonomy table, and phylogenetic tree. Supplementary materials will show how to import Bracken shotgun metagenomic abundance data and generic abundance data frames into a phyloseq object.

[**Intro**](#datasetandworkflowchap)
[{style="height:150px"}](#datasetandworkflowchap)
[**Set-up**](#setupchap)
[{style="height:150px"}](#setupchap)
[**Data preparation**](#preprocess_section)
[{style="height:150px"}](#preprocess_section)
[**Taxonomy relative abundance**](#taxa_relabund_chap)
[{style="height:150px"}](#taxa_relabund_chap)
[**Rarefaction**](#rarefaction_chap)
[{style="height:150px"}](#rarefaction_chap)
[**Beta diversity**](#beta_chap)
[{style="height:150px"}](#beta_chap)
[**Summary**](#sumchap)
[{style="height:150px; border-radius:15px; background:white"}](#sumchap)

The sessions will start with a brief presentation followed by self-paced computer practicals guided by this online interactive book. This book contains theory and practice code. This will be reinforced with multiple choice questions that will recap concepts and aid in interpretation of results.

At the end of the course learners will be able to:

- Import QIIME2 artifacts into a phyloseq object.
- Summarise the abundance and taxonomy contents of a phyloseq object
- Preprocess the abundance and taxonomy tables. This will include transforming sample counts, and subsetting samples & taxonomies.
- Understand the grammar of graphics (ggplot2) used by phyloseq and related packages.
- Carry out alpha & beta diversity, and biomarker detection with the phyloseq object.
- Produce and customise publication quality plots.
- Run statistical analysis.

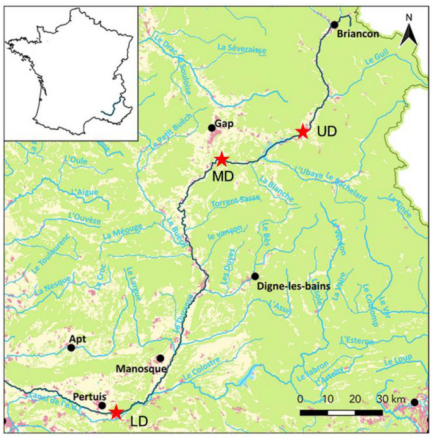
There are supplemental materials including importing other types of data into phyloseq, and the use of R plotly to produce interactive HTML based plots.

Table of contents

Chapter 2

Dataset & workflow

2.1 Dataset



In this tutorial we will be using a 16S metabarcoding dataset derived from surface water from the Durance River in the south-east of France. Two major comparisons were carried out in combination with each other.

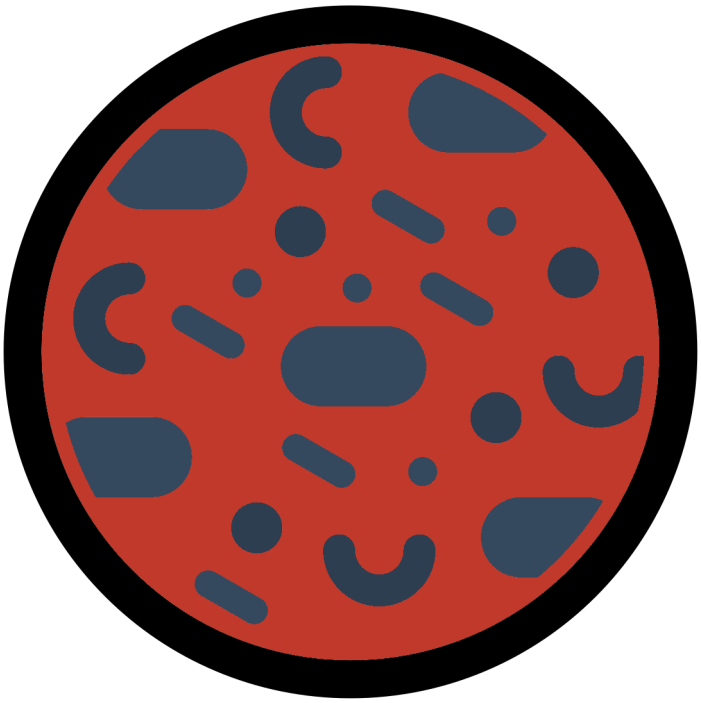
Link to paper

2.1.1 Sites

Three different sites were chosen on the Durance River. These three sites were representative of an anthropisation (transformation of land by humans) gradient along a river stream. These sites were:

- **Upper Durance sampling site (UD)**: Alpine part of the river with little/no anthropisation.
- **Middle Durance sampling site (MD)**: Upper part of agricultural land dominated by apple and pear production.
- **Lower Durance sampling site (LD)**: Lower part of agricultural land with intensive production of fruits, cereals, and vegetables.

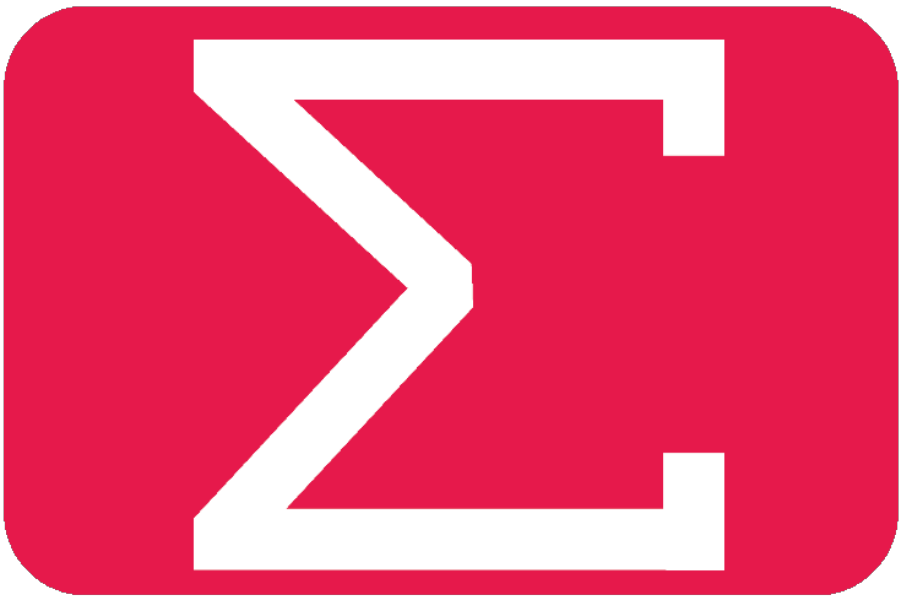
2.1.2 Culture media



Surface water was sampled and different culture media were used to produce bacterial lawns for each site. The media used were:

- **Environmental sample (ENV)**: No media used, frozen at -20°C.
- **TSA 10%** incubated at 28°C for 2 days.
- **KBC** incubated at 28°C for 2 days.
- **CVP** incubated at 28°C for 3 days.

2.1.3 Summary & questions

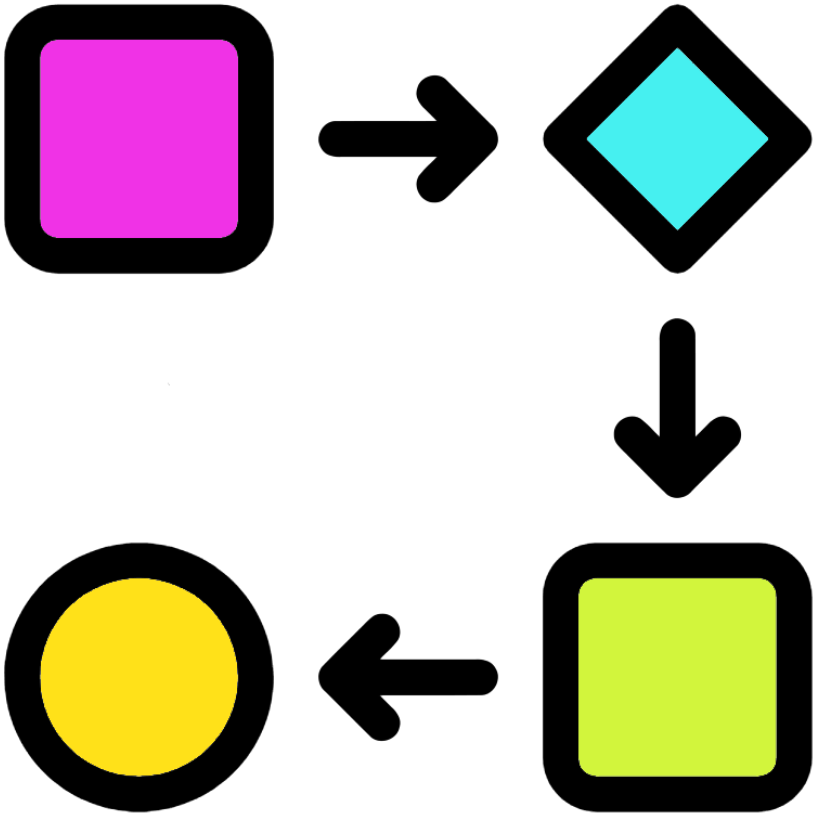


Each sample and media combination was produced in replicates of three giving a total of 36 samples (3 X 4 X 3 = 36). The three replicates were cultured on three different plates with the same media. An ASV table, taxonomy table, and phylogenetic tree were produced with QIIME2 and DADA2.

With this data we can ask and investigate the following questions:

- How do the bacterial communities change across the anthropisation gradient?
- Is there a difference in the replicates of one site and media combination? I.e. do any of the media produce inconsistent profiles?
- Is there more difference between the sites or the media used?
- Do the media samples differ from the ENV samples? If so, how?

2.2 Workflow



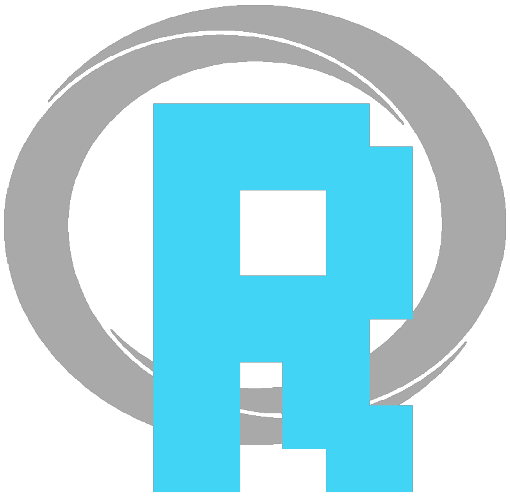
1. Import: Import QIIME2 artifacts into a `phyloseq` object with `qiime2R`.
2. Summarisations: Check our `phyloseq` object with summarisations.
3. Minimum depth: Determine the minimum depth we should use and remove samples with lower depth.
4. Taxanomic relative abundance: Create taxonomic relative abundance tables.
5. Taxa plots: Produce heat maps and bar plots of taxa relative abundances.
6. Family and genus: Using the last step to produce family and genus based taxa plots.
7. Rarefaction: Carry out sample depth normalisation with rarefactions. This will be used for alpha and beta diversity analysis.
8. Alpha diversity: Carry out alpha diversity analysis through plots and statistics.
9. Beta diversity: Carry out beta diversity analysis through plots and statis-

tics.

10. Differenital abundance anlaysis: Detect biomarkers compared to a reference group with ANCOM.

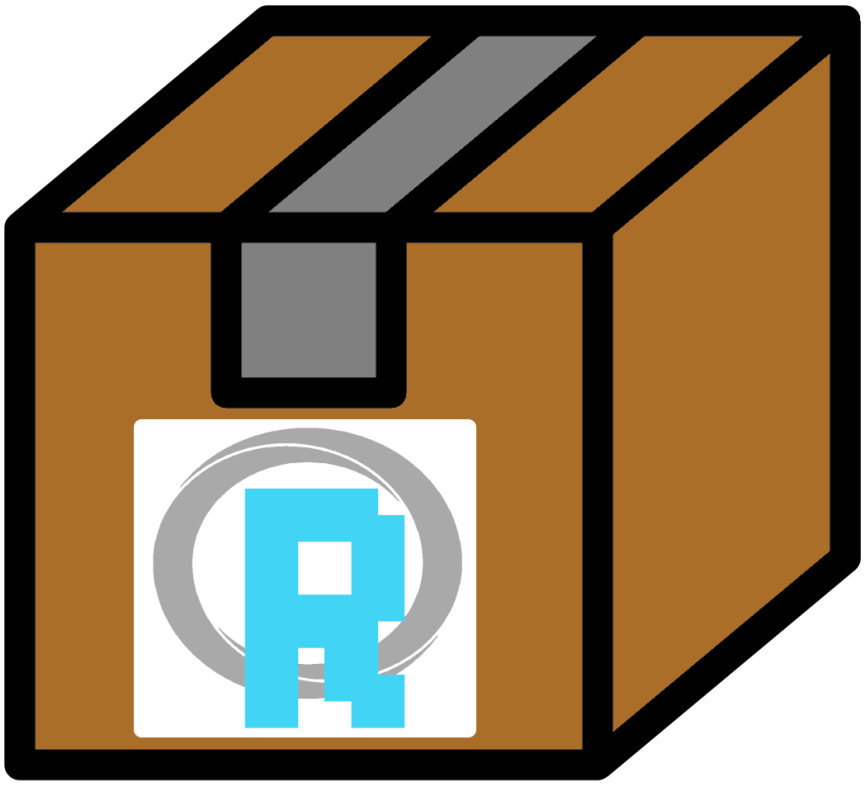
Chapter 3

R Packages



During this workshop we will use various R packages with their own intricacies. Before going into analysis we'll introduce you to some of these important concepts.

3.1 R packages/libraries



R packages/libraries contain additional functions, data and code for analysing, manipulating and plotting different types of data. Many common packages will be installed as default when you install R. Other more specialised packages, such as the `ggplot2` package, must be installed by the user.

Packages found on The Comprehensive R Archive Network (CRAN), R's central software repository, can be installed using the following command.

```
install.packages("package_name")
```

Every time you reload R you will need to load the packages you require if they are not installed in R by default. To do this type:

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

I generally have a list of `library()` functions at the top of my R scripts (`.R` files) for all the packages I use in the script.

Throughout this course you will get a lot of practice installing and loading various packages.

R package or R Library?

R packages are a collection of R functions, data, and compiled code. You can install these into a directory on your computer.

An R library is a directory containing a R package.

Because of this, the terms R package and R library may be used synonymously. We will use the term package in this workshop.

As we will be using a lot of packages we shall use double colons (`::`) to specify which package each function belongs to, unless the function is from base R. For example if we use the function `summarize_phyloseq()` from the package `microbiome` we would type the function like below:

Note: Do not run the below command.

```
microbiome::summarize_phyloseq()
```

This convention has 2 benefits:

- We can easily tell which R package each function comes from.
  - This is useful for your future coding where you may copy some, but not all, commands from one script to another. You will therefore know which packages you will need to load.
  - If you need some more documentation about a function you will know what package to look up.
  - Writing your methods will be a lot easier.
- Different packages may have functions with the same name. Specifying the package will ensure you are using the correct function.

3.2 The grammar of graphics

During this course we will be using the grammar of graphics coding approach. This approach is implemented by the R package `ggplot2` to create visualisations such as bar charts, box plots, ordination plots etc. In turn `ggplot2` is used by

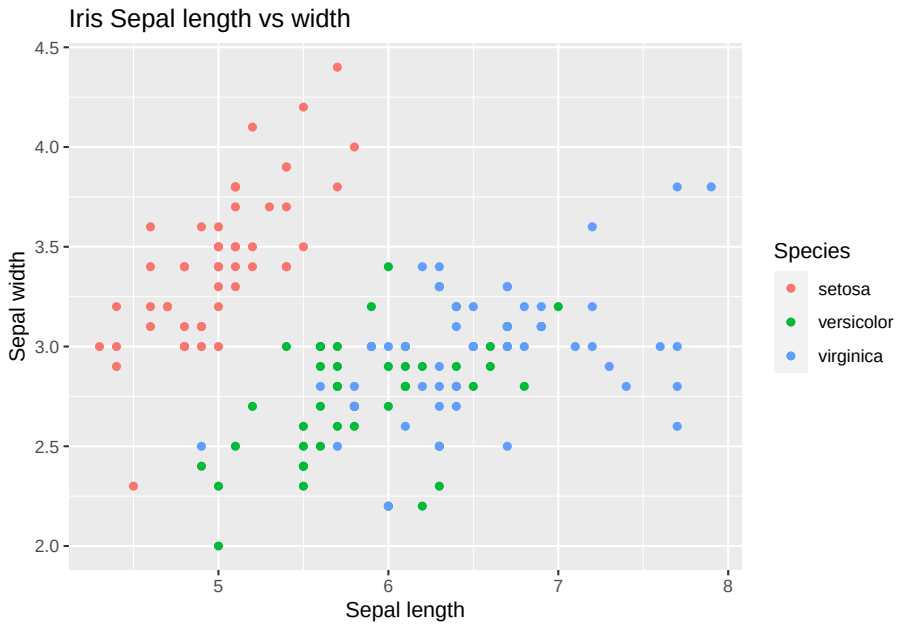
a host of other packages, some of which we will be using. Although `ggplot2` is R code, its structure is very different and it takes effort to learn. Thankfully, `ggplot2` is very powerful and flexible, and it produces very professional and clean plots.

We will use the `iris` dataset (inbuilt into R) to show an example of `ggplot2` code and its visualisation output. You don't need to run the below code.

Note: If you would like to see the contents of the `iris` dataset you can run the command `View(iris)` in your R instance later.

```
#Load library
library(ggplot2)

#Create new ggplot2 object using iris dataset
ggplot2::ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, colour=Species)) +
  #Make the object a scatter plot
  ggplot2::geom_point() +
  #Add plot tile
  ggplot2::ggtitle("Iris Sepal length vs width") +
  #Set x and y axis label names
  ggplot2::labs(x = "Sepal length", y = "Sepal width")
```



We will not learn `ggplot2` specifically during this course. However, the structure of creating an object will be used. In the above case the initial object was built

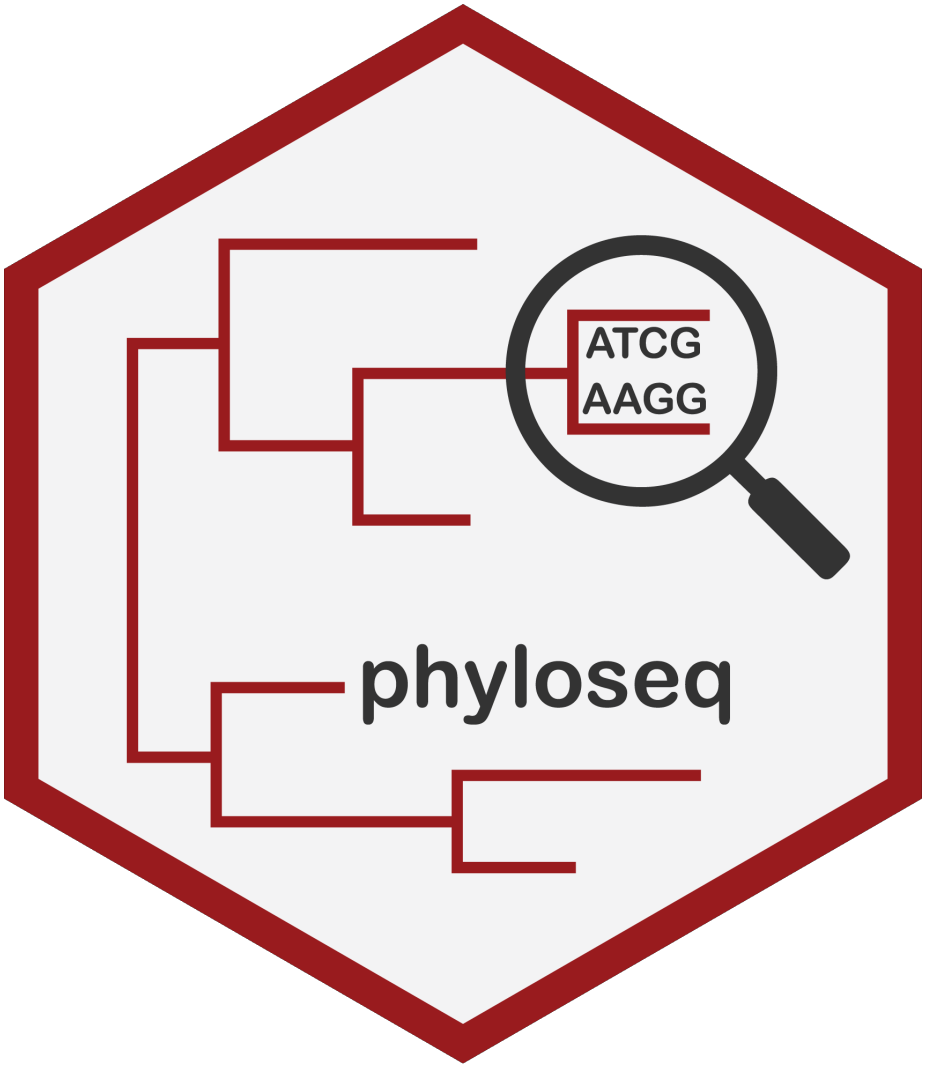
with `ggplot`. Subsequently additions and edits were carried out with `+` and various other functions.

An important concept of the grammar of graphics is aesthetics. Aesthetics are the parts of a graphic/plot. In the above command we set the aesthetics with the function `aes()` within the `ggplot()` function. The X aesthetic (i.e. what values are assigned to the x axis) was set as the Sepal length values from the column `Sepal.Length` of the dataframe `iris`. In turn the Y axis values are set to the Sepal width and the colouring of the points are set to the Species.

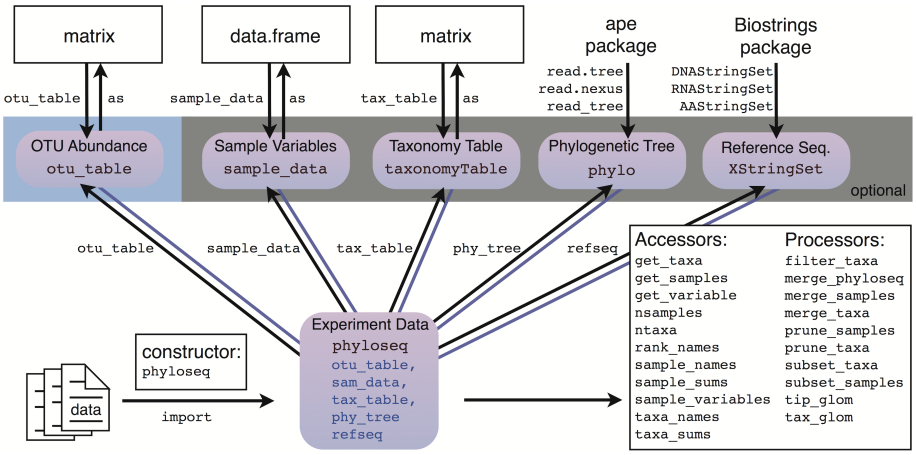
That was a quick introduction to the grammar of graphics. We will be using this to create visualisations with a `phyloseq` object using various R packages specifically designed for community abundance data within `phyloseq` objects.

For more resources on `ggplot2` please see the appendix of this book.

3.3 phyloseq



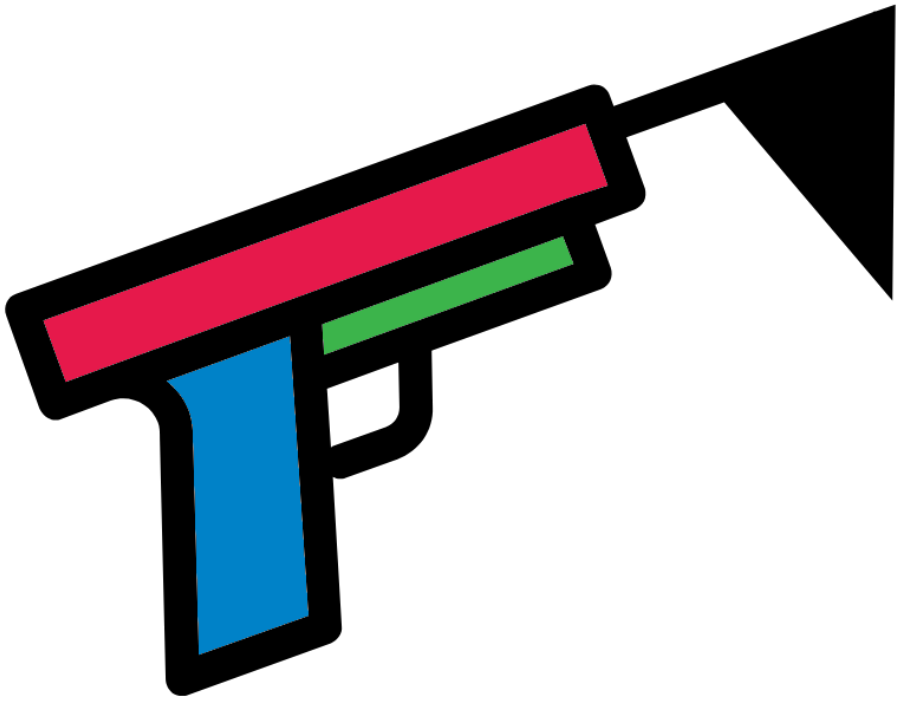
In this book we will be working with `phyloseq` objects to preprocess our dataset, create visualisations, and carry out statistical analyses. This is a very popular object type for community abundance datasets as it contains the abundance table, metadata, and taxonomy table in one object, optionally containing the phylogenetic tree and reference sequences if wanted/required.



For more info on `phyloseq` and associated packages please see the appendix.

Chapter 4

Set-up

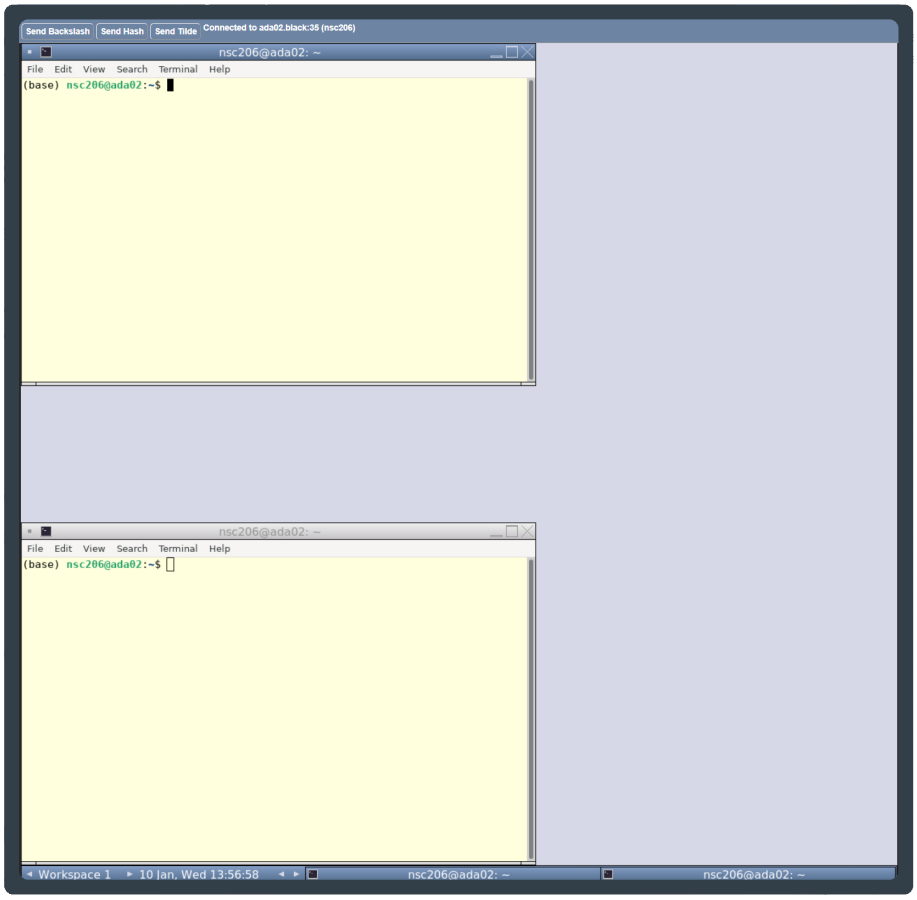


Prior to any analysis we need to setup our environment in the webVNC.

4.1 Logon instructions

For this workshop we will be using Virtual Network Computing (VNC). Connect to the VNC with a browser by using the webVNC link you were sent.

You will now be in a logged-in Linux VNC desktop. You will see something as below (there may be only one terminal which is fine). If you do not see something similar please ask for assistance.

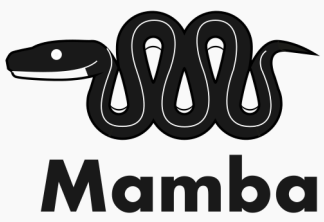


If the VNC is taking up too much/little space of your browser you can use the zoom of your browser to adjust the size. You will most likely need to use your browser's tool bar to accomplish this. Ensure you can see the grey borders.

These instructions will not work outside of this workshop. If you would like to install your own Linux OS on your desktop or laptop we would recommend Mint Linux

The following link is a guide to install Mint Linux:
https://linuxmint-installation-guide.readthedocs.io/en/latest/

4.2 Mamba



This workshop requires a lot of packages. These all can be difficult to install with R. Instead we have used Mamba forge to install R, its packages, and Jupyter-notebook (more info below). To learn more about Mamba-forge and how to create your own environment please see the appendix.

To set-up your environment for this workshop please run the following code (you must include the full stop and space at the front of the command).

```
. usercommunity
```

You will have successfully activated the environment if you now see `(r_community)` at the start of your command prompt. This indicates you are now in the mamba environment called `r_community` created by the instructor.

If you are interested in the use script you can look at its contents.

```
less /usr/local/bin/usercommunity
```

Tip: press `q` to quit `less`.

For more about mamba and how to create your own `r_community` environment please see the appendix

Chapter 5

Jupyter