# BIO445 Quantitative Life Sciences

Virulence Evolution - Exercise Sheet

Marco Labarile, marco.labarile@uzh.ch

2022-12-21

## R toolbox: lapply, sapply and match

In this module we will use the functions `lapply` and `sapply`. To make you familiar with them, generate the following list in R, read up the documentation of `lapply` and `sapply` and then complete the following tasks.

```
example <- list(1:5, rep(5, 10), 10:5, c(3, 6))
help(lapply)
help(sapply)
```

Using `lapply` and/or `sapply`, generate the following objects:

1. A vector containing the length of each entry of `example`.
2. A vector containing the mean of each entry of `example`.
3. A list containing the sum of each entry of `example`.
4. A list containing each entry of `example` plus 1.

Additionally, we will make use of the R function `match`. To illustrate how `match` works, we will use the alphabet (stored in R as `letters`).

```
help(match)
letters  # the whole alphabet as a character vector
letters[1:3]  # the letters a, b, and c
```

5. Using `match`, find the position of "s" in `letters`.
6. At which position is "s" after randomly shuffling the alphabet (using the function `sample`)?
7. What is the mean position of the character "s" after shuffling the alphabet 1000 times?

For this module we need the R package *ape*. If it isn't already installed, install it as usual with `install.packages`.

```
install.packages("ape")
require(ape)
```

## Problem 1: Evolution of HIV virulence

In this exercise we want to simulate the evolution of HIV virulence. After a period of acute HIV infection, the viral load which stabilizes is called set-point viral load (measured in RNA copies per ml). It has been shown that the set-point viral load is strongly associated with the time to AIDS development of the patient, which is highly variable (around 2-20 years).
We will investigate whether it is possible to explain the observed distribution of set-point viral loads as the combined effect of the following forces:

- **Natural selection:** Different HIV strains have different transmission potential (depending on their set-point viral load). Does HIV try to maximize opportunities for onwards transmission?

- **Mutation:** The HIV genome mutates along its transmission chain and therefore the virulence of different strains in a population will differ.
- **(Environmental) Noise:** We expect that even the same viral strain will produce different levels of virulence in different patients, e.g., as the result of host genetics or other unaccounted for factors.

We want to answer this question by making individual-based simulations over many generations of HIV-infected individuals. For that purpose, we use data of the paper *Variation in HIV-1 set-point viral load: Epidemiological analysis and an evolutionary hypothesis* by Fraser et al.

### 1.1

Read in the file `vl_tr_data.csv`. In the first column you find viral load values and in the second column you have transmission rates. Why would it be useful to log10-tranform the viral load values? To study the relationship between set-point viral load and transmission rates we will use linear interpolation. This tool is often used to approximate some function and implemented as `approxfun` in R.

```
vl_tr_fun <- approxfun(vl_tr$logvl, vl_tr$tr, method = "linear",
    rule = 2)
```

Next, try to understand what we are plotting with the following command:

```
plot(seq(1, 7, by = 0.1), sapply(seq(1, 7, by = 0.1), vl_tr_fun),
    type = "l", lty = 2, xlim = c(3, 6), xlab = "log10 viral load",
    ylab = "transmission rate")
```

### 1.2

Read in the file `vl_aids_data.csv`, which contains a column with viral load values and one with time to AIDS values. Use the same tools as in 1.1. to visualize the association between the viral load and time to AIDS.

### 1.3

We approximate the viral fitness as the product of (1) transmission rate and (2) mean time until AIDS progression. Write a third function that does that and then plot the association between viral load and viral fitness. For which viral load does the viral fitness peak?

### 1.4

Finally, we start with our simulations.

We assume a population size of 1,000 and model 500 generations of this population (feel free to adjust these parameters later).

```
population_size <- 1000
number_generations <- 500
```

We model the viral load as a phenotype that is the sum of a genetic and an environmental component. The genetic component undergoes mutations (modeled as normally distributed) and is then passed on to the offspring. In addition, the environmental component is added to the offspring to get the actual (realized) phenotype.

```
sigma_environmental <- 0.8
sigma_mutation <- 0.1
```

Prepare two matrices that you will fill with the genetic and realized values of set-point viral load produced during the entire simulation.

```
population_genetic <- ??
population_realized <- ??
```

Assume that all individuals in the first generation have the initial log10-transformed viral load of 3.

```
initial_vl <- 3
population_genetic[, 1] <- ??
```

Now we are ready to actually simulate: Each k in our for-loop represents one generation. Complete the ?? according to the comments.

```
#for loop over the number of generations
for(k in 1:(?? - 1)){
  #extract the genetic component of virus load in the kth generation of the matrix
  genetic_vl <- ??

  #add normal distributed environmental term to get the realized virus load
  population_realized[, k] <- ??

  #get the fitness of the kth population
  fitness <- sapply(??, ??)

  #get the number of offspring by using a multinomial distribution
  #the population size remains constant but fitter individuals are
  #more likely to have offspring
  nr_offspring <- rmultinom(1, population_size, fitness)

  #genetic component is passed on to the offspring
  genetic_vl_unmut <- unlist(sapply(??, FUN =
                                      function(x) { rep(genetic_vl[x], nr_offspring[x]) } ))

  #add mutations to the viral component, approximated by normal distribution,
  #save it as genetic components of next generation
  population_genetic[, k + 1] <- ??
}
```

### 1.5

Having the simulated data ready, we want to discuss, interpret and visualize this data.

Make a plot to show how the population mean viral load behaves over time. What do you observe? What do you observe only considering the first 50 generations?

To illustrate how the virulence of HIV evolves over time we want to combine our plot from 1.3. with the distribution of the viral load in different generations.

```
#plot from 1.3 (viral load vs. viral fitness)
plot(??)

#nice colours: start with blue colors and go more and more towards coral coloured ones
palette <- colorRampPalette(c("darkblue", "coral"))

#which generation do we want to show?
seq <- seq(10, 80, by = 10)
palette <- palette(length(seq))

for(k in seq){
```

```
  lines(density(??), col = palette[k/10])
}

legend("topright", col = palette, legend = paste("gen", seq), lwd = 1)
```

Discuss this finding! Thinking of the real HIV epidemic - in which generation would we be nowadays?

# Problem 2: Heritability of HIV set-point viral load

In this exercise we want to quantify the contribution of HIV viral genetic factors to its set-point viral load, i.e. the heritability of set-point viral load.
You will be provided with a phylogenetic tree, which shows the evolutionary relationship of the HIV strains isolated from each patient based on their genetic sequence. The tips of the tree are marked with the set-point viral load that corresponds to patients.

## 2.1

To use parent-offspring regression, we first need to find transmission pairs in the phylogeny. How would you choose those pairs? Discuss and mark them in the figure.

## 2.2

Read the tree into R by running the following code:

```
tree_spvl <- read.tree(file = "tree_spvl")
```

Think about how to implement a function in R that extracts transmission pairs for you. Find out what the `ape::extract.clade` function does.

## 2.3.*

Complete the ?? in the following code that extracts transmission pairs. If you want to skip this, have a look at the functions at the bottom of the starting script, in the section "Functions."

```
#This function gives the label of a node pointing to the tip
find_node <- function(??, ??) {
  tip <- match(id, tree$tip.label)
  node <- tree$edge[match(as.numeric(??), tree$edge[, 2]), ][1]

  if(is.na(node) == "FALSE") {
    return(as.numeric(node))
  }
}

#This function returns c(NA,NA) if this patient (tiplabel) is not in a transmission
#pair and the name of the tiplabels if he/she is in a transmission pair
#(so the tiplabel of the patient and the other member of the pair)
find_pair <- function(??, ??) {
  clade <- extract.clade(??)

  if (length(clade$tip.label) != ??) {
    return (c(NA, NA))
  } else {
    return (??)
  }
```

```
}

#This function returns all transmission pairs
transmission_pairs <- function(??) {
  pair <- matrix(NA, length(tree$tip.label), 2)
  pair <- t(sapply(tree$tip.label, FUN = function(a) { find_pair(??) } ))
  pair <- pair[which(!is.na(??)), ]
  order <- function(row) { paste0(min(row), "_", max(row)) }
  pair <- unique(sapply(pair, order))
  pair <- lapply(pair, FUN = function(a) { unlist(strsplit(a, "_")) } )
  return(??)
}

pairs <- ??
```

### 2.4

To get the heritability of HIV set-point viral load, perform a linear regression with the first element of each entry in your `spvl` list as parent and the second element as offspring.

```
parent <- sapply(pairs, FUN = function(a) { unlist(a)[??] } )
offspring <- sapply(pairs, FUN = function(a) { unlist(a)[??] } )
??
```

What conclusions can you draw about the heritability of set-point viral load?

### 2.5*

Modify the function(s) such that only transmission pairs with a genetic distance below a threshold x (for example 2.5%) are returned. Does the choice of the distance threshold influence the heritability?