# Data preparation and first steps

Module: Plant Genetic Resources (3502-470)

Karl Schmid, Katharina Böndel, University of Hohenheim

Summer Term 2024

## Background information about RMarkdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Mathematical symbols are written in the LaTeX syntax, e.g, nucleotide diversity pi: $\pi$

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can decide whether you want a pdf, an html or a word document.

## Working in the Rstudio cloud

Although you can always download R and Rstudio to your local computer and both may be are already installed on the computers at the university, we would recommend you to use the Rstudio Cloud (https://rstudio.cloud/) for this module. This way you can access your data and code from everywhere and continue your work. Furthermore, you will encounter less problems with package installation in the cloud then on your local system.

Once you have created an account, you can directly open a workspace on the left ("+ New Space") and create a project there. To upload files (data or code) go to the "Files" tab in the right bottom panel and chose "Upload".

## Packages in R

We will need various R packages throughout the module. To install an R package you have to use the `install.packages()` command, e.g. to install the R package `vcfR` you use `install.packages("vcfR")`. An R package is then loaded with `library()`, e.g. `library(vcfR)`. If you need help or want to see the documentation of a package just type in `?vcfR` and if you type in `library(help="vcfR")` you get a list with all commands from the package.

## The data set

We will work with one data set throughout this module. This data set contains several teosinte (*Zea mays parviglumis*) individuals, maize landraces and maize improved varieties. The data spans a 3Mb region on chromosome 1. The data is stored in a file format called `vcf`: `zea.vcf`. Next to this file, there are three text files containing the sample names of each of the three groups: `teosinte.csv`, `landraces.csv` and `improved_varieties.csv`. Make sure that all four files are uploaded to your Rstudio Cloud project. If you decide to work on a local system, make sure to set your working directory accordingly and/or to always give

the path to the input files. Otherwise you will get error messages saying that the file is not found or that the connection cannot be opened.

## R packages for this computer lab

For this computer lab we will need the R packages:

- `vcfR` to read the vcf
- `pegas` to perform the analyses

Install and load both packages:

```
install.packages("vcfR")
library(vcfR)
install.packages("pegas")
library(pegas)
```

## Prepare the data

Before we can start with any analysis, we first need to prepare the data.

We load the vcf and then write the data into an `DNAbin` object (to learn about this class of objects, type `?DNAbin`):

```
zea_vcf <- read.vcfR("zea.vcf")
zea <- vcfR2DNAbin(zea_vcf,unphased_as_NA = FALSE,extract.haps = TRUE)
```

Have a look at the screen output; it gives you some information about number of variants, number of individuals, etc.

Next, we want to keep only variants that are useful for our analyses, that is we want to exclude any positions with missing data (NA) or variants that have more than two states, i.e. are not biallelic. (*Please note that filtering criteria always depend on the analyses you want to perform. For some analyses you may need to keep all variants or even invariant sites. This also means that a thorough understanding of the type of analysis and method is essential to perform good data analyses.*) For this, we first write a function, then run this function to get this information for each position and finally exclude all positions that do not match the criteria:

```
# Function to only retain biallelic SNPs with no missing data
only_biall_noNA <- function(v){v <- as.character(v)
                           includeSNP <- all(v %in% c("a","c","t","g"))
                           includeSNP <- includeSNP & (length(unique(v))==2)
                           return(includeSNP)}

# whichSNPs contains for each SNP whether it is biallelic and has no missing data
whichSNPs <- rep(FALSE,ncol(zea))
for (i in 1:ncol(zea)){
whichSNPs[i] <- only_biall_noNA(zea[,i])
}

# Only keep such SNPs
zea_snps <- zea[,whichSNPs]
```

We will now save this `DNAbin` object to load it directly in the future computer labs.

```
save(zea_snps,file="zea_snps.RData")
```

Since we have teosinte, landraces and improved varieties in our data set, we also need to define accession groups.

We can first look at the names of the individual accessions:

```
acc_names <- rownames(zea_snps)
acc_names
```

Then we read in our text files which contain the accession names of each group and define accession lists that we can later use to specifically analyse only the accessions from one of the lists:

```
# all teosinte individuals
teo_list <- unname(unlist(read.table("teosinte.csv",as.is = TRUE)))
teo_list
teo_list2 <- c(paste0(teo_list,"_0"),paste0(teo_list,"_1"))
teosinte_acc <- which(acc_names %in% teo_list2)

# all landraces
landraces_list <- unname(unlist(read.table("landraces.csv",as.is = TRUE)))
landraces_list
landraces_list2 <- c(paste0(landraces_list,"_0"),paste0(landraces_list,"_1"))
landraces_acc <- which(acc_names %in% landraces_list2)

# all improved varieties
varieties_list <- unname(unlist(read.table("improved_varieties.csv",as.is = TRUE)))
varieties_list
varieties_list2 <- c(paste0(varieties_list,"_0"),paste0(varieties_list,"_1"))
varieties_acc <- which(acc_names %in% varieties_list2)
```

We also save the accession lists so that we can directly load them in the future:

```
save(teosinte_acc,file="teosinte_acc.Rdata")
save(landraces_acc,file="landraces_acc.Rdata")
save(varieties_acc,file="varieties_acc.Rdata")
```

## A first look at the data set

We can visually display the data with the following commands:

```
image.DNAbin(zea_snps[teosinte_acc,],main="teosinte indviduals")
image.DNAbin(zea_snps[landraces_acc,],main="landraces")
image.DNAbin(zea_snps[varieties_acc,],main="improved varieties")
```

**Q**: Try to understand what you see: What do the colours show? What is on the y-axis and what is on the x-axis?

**A**:

**Q**: Compare the three data sets: Can you see any differences? What could they mean?

**A**:

To better compare them, it helps to plot them together into one graph. This can be achieved with the `par(mfrow=c())` command:

```
par(mfrow=c(3,1)) # this allows to plot the three graphs together
                  # and one below the other: 3 rows and 1 column of graphs
image.DNAbin(zea_snps[teosinte_acc,],main="teosinte indviduals")
image.DNAbin(zea_snps[landraces_acc,],main="landraces")
```

```
image.DNAbin(zea_snps[varieties_acc,],main="improved varieties")
par(mfrow=c(1,1)) # this "resets" the previous par command
```

*Note: you may need to make the panel that shows the plots bigger in order to see them. Alternatively, you can also save them in a pdf and then open the pdf. This you can do by running* `pdf("zea-data.pdf")` *before the code chunk and* `dev.off()` *after the code chunk.*