dartR Workshop - GSA

Bernd Gruber & Arthur Georges

2018-06-20

Foreword

What it the workshop about?

In this workshop we provide an overview on the use of a recently developed R-package (dartR) that aims to integrate as many as possible ways to analyse SNP data sets. We will cover the following topics:

Overview (indicated times in brackets)

0. Learning R on (the quickest intro possible) (9:00-10:00) - Bernd

   - Rstudio
   - packages
   - Objects
   - vectors
   - matrices
   - plotting
   - indexing

1. Preparing data sets (10:00 - 10:30) - Bernd

   - loading data sets into R (DArT format)
   - how to load other formats

2. The idea of the a genlight object (10:45 - 11:00) - Bernd/Arthur

   - explore a genlight object
   - quality filter your data
   - subset/recode your data set

3. Population structure (11:00 - 11:45) - Arthur

   - PCoA
   - phylogenetic trees
   - fixed differences

4. Population assignment (11:45 - 12:15) - Arthur

5. Landscape genetics (12:45 - 13:15) - Bernd

   - isolation by distance
   - landscape resistance

6. Export data set to other formats (13:15-13:45) - Bernd

- how to send a genlight objectto a friend
- FASTA
- STRUCTURE, fastSTRUCTURE, NewHybrid

The format of the workshop will be a mixture between short lectures on the different topics to explain the idea of an approach and the type of analysis followed by computer exercises how to implement and interpret such an analysis. To avoid setting up computers we will use our Rstudio server, with the advantage that everyone will be on the same page. Data sets can be downloaded from here [`http:github/dartRworkshop/data`]

Login details:

There are two logins, the first to log into the University computer, the second to log into Rstudo on the server.

1. Univeristy login: user: ucstaff\ucvisitor14 pw: ucvisitor
2. Rstudio login. [Open a web browser (chrome, firefox)]

- Goto: dungog.win.canberra.edu.au:8787
- user: guestxyz [your guest number]
- pw: guestxyz [your guest number]

You should now see and Rstudio window such as the one below:

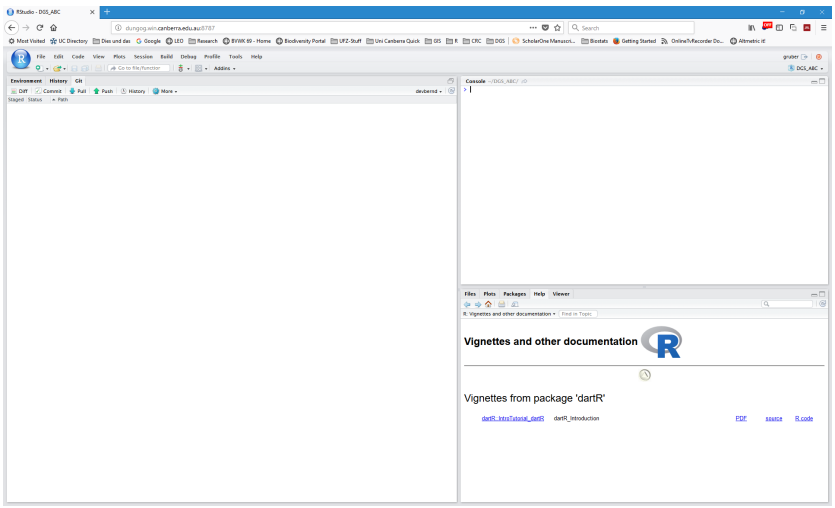


Figure 1: Rstudio

0. Learning R on 1-2 pages (9:00-10:00)

Obviously we cannot teach you R on a page. Rather this is a very quick refresher of a very small selection of R concepts, which we will

need during the workshop. A good starting points are the cheatshets linked from Rstudio (Help->Cheatsheats), but feel free to google for some excellent and free introductions to R.

Rstudio

There are four windows (panes in Rstudio).

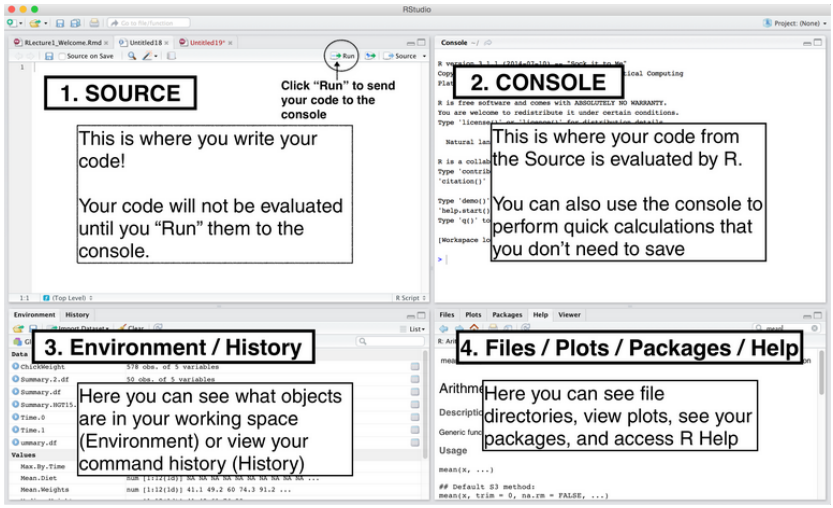


Figure 2: Windows in R studio

Packages

Before you start an analysis in R you need to load additional functions. The package we will use today is called dartR. The first code you should type in your source window (after you created a new file Shift-Control-N) is:

```
library(dartR)
```

```
## Loading required package: adegenet
```

```
## Loading required package: ade4
```

```
##
##    /// adegenet 2.1.1 is loaded ////////////
##
##    > overview: '?adegenet'
##    > tutorials/doc/questions: 'adegenetWeb()'
##    > bug reports/feature requests: adegenetIssues()
```

This loads the dartR package into your session and all functions included in the package can now be used. Be aware once back at home

you need to download and install dartR on your computer once before you can use it.[1]

[1] For information how to install dartR please refer to:`https://github.com/green-striped-gecko/dartR`

Objects and vectors

Data are stored in objects in R. Those objects have names and we can create them:

```
mydata <- c(6, 7, 3, 8)
```

And look into the content using the name and send it to the console:

```
mydata
```

```
## [1] 6 7 3 8
```

There are simple standard accessor functions you can use on every object.

```
str(mydata)
```

```
##  num [1:4] 6 7 3 8
```

```
class(mydata)
```

```
## [1] "numeric"
```

Often there are some additional function that work with certain types of objects. For example mydata is a vector (a one dimensional array/container) that holds numeric data. For this kind of data there are a lot of functions, such as[2]:

[2] When you learn a new package you often do not know the name of functions. Here a good start is to look at so-called tutorials for packages. For example dartR has a vignette that lists (almost) all available functions: Type: `browseVignettes("dartR")` into the console

```
mean(mydata)
```

```
## [1] 6
```

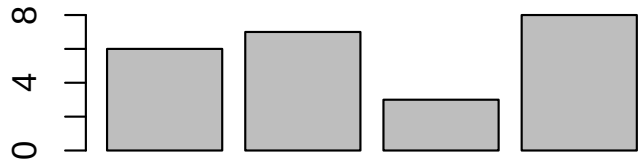
```
summary(mydata)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.
##    3.00    5.25    6.50    6.00    7.25
##    Max.
##    8.00
```

```
length(mydata)
```

```
## [1] 4
```

```
barplot(mydata)
```



matrices/data.frame

A matrix is basically the R equivalent of a table in Excel and it can store two dimnensional data sets. En example for this kind of data is the built-in data set on iris flowers (Fisher/Anderson 1936).

```
iris
```

Typing the name shows you the content (a long table, not shown here). There are several ways to summarise the data set using built-in functions for data.frames.

```
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"
## [3] "Petal.Length" "Petal.Width"
## [5] "Species"
```

```
nrow(iris)
```

```
## [1] 150
```

```
ncol(iris)
```

```
## [1] 5
```

```
dim(iris)
```

```
## [1] 150   5
```

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width
##  Min.   :4.300   Min.   :2.000
##  1st Qu.:5.100   1st Qu.:2.800
##  Median :5.800   Median :3.000
##  Mean   :5.843   Mean   :3.057
##  3rd Qu.:6.400   3rd Qu.:3.300
##  Max.   :7.900   Max.   :4.400
##   Petal.Length    Petal.Width
##  Min.   :1.000   Min.   :0.100
##  1st Qu.:1.600   1st Qu.:0.300
##  Median :4.350   Median :1.300
##  Mean   :3.758   Mean   :1.199
##  3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

indexing (finding subsets)

Being able to create subsets in R opens the avenue to a very powerful way to analyse your data. E.g. being able to run the same analysis for males and females seperately is often very simple in R, once you scripted your analysis. R achieves that via the indexing function "[]". A matrix/data.frame consists of rows and columns and we can use the indexing function to identify certain columns and rows.

```
iris[3, 4]  #returns the value in row 3, column 4
```

```
## [1] 0.2
```

```
iris[1:3, 4]  #returns the value from row 1 to 3 or column 4 (=a vector)
```

```
## [1] 0.2 0.2 0.2
```

```
iris[1:3, 4:5]  #returns values from row 1 to 3 and column 4 and 5 (=a matrix)
```

```
##   Petal.Width Species
## 1         0.2  setosa
## 2         0.2  setosa
## 3         0.2  setosa
```

We can now use certain colums to find subsets of interest. E.g. if you want to calculate the mean Petal.Length for 'setosa' we have to do two steps. First find all the rows that have 'setosa' in the Species column and then use that as an index for the column Petal.Length to calculate the mean.

```
### Step 1a: Finding the Species column
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"
## [3] "Petal.Length" "Petal.Width"
## [5] "Species"
```

```
### Step 1b: create an index
index <- iris$Species == "setosa"

### Step 2: calculate the mean
mean(iris$Petal.Length[index])
```

```
## [1] 1.462
```

The same result in a "shorter way". We can use the $ function to find a column of a matrix or we can use numbers/names to identify columns.

```
mean(iris[iris$Species == "setosa", "Petal.Length"])
```

```
## [1] 1.462
```

```
# only numbers
mean(iris[1:50, 3])
```

```
## [1] 1.462
```

Admittedly it takes some time to get used to it, but once the concept is understood it is very powerful. The good news is that dartR provides some help functions to subset data set, hence at the beginning subsetting using the '[]' function is not necessary.]

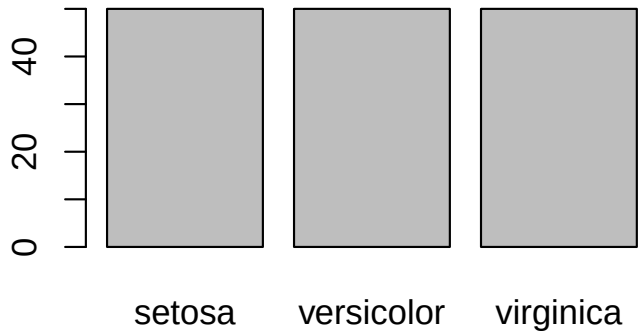
Plotting and summarising

There hundreds of different ways

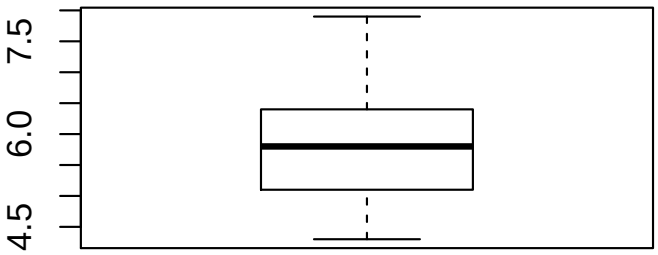
```
table(iris$Species)
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

```
barplot(table(iris$Species))
```




```
boxplot(iris$Sepal.Length)
```



Today we will use a quite specialised type of object which are called genlight objects. And here comes the first task. Access the built-in data set `testset.gl` and find out the following:

Question

1. What type of data set is `testset.gl`?
2. Can you find the help page for the data set.
3. Type the name and try to understand the "structure" of this data type
4. Try the plot function on this data set.
5. There are additional accessor functions such as `indNames(), nLoc(), nInd(), ploidy()`
6. Try to use table function on pop(testset.gl).

- How many populations are there?
- Can you produce a barplot on the number of individuals per population?

7. **Extra: Can you subset the first four individuals and the first 10 loci and plot them?**

1. Preparing data sets (10:00 - 10:30)

- `loading data sets into R (DArT format)`

- how to load other formats

[Mainly from vignette....

- explain the data format from dartR
- Locus meta data
- Individual meta data

2. The idea of the a genlight object (10:45 - 11:00)

- explore a genlight object
- quality filter your data
- subset/recode your data set

3. Population structure (11:00 - 11:45)

- PCoA
- phylogenetic trees
- [snapclust]
- fixed differences

4. Population assignment (11:45 - 12:15)

5. Landscape genetics (12:45 - 13:15)

- isolation by distance
- landscape resistance

6. Export data set to other formats (13:15-13:45)

- how to send a genlight objectto a friend
- FASTA
- STRUCTURE, fastSTRUCTURE, NewHybrid