A (Very) Short Introduction to R for Wet Lab Scientists

Kieran O'Neill

What is R, and Why Should I Use It?

The Basics

R is a versatile, open source programming language that was specifically designed for data analysis. As such R is extremely useful both for statistics and data science. Inspired by the programming language S.

- ▶ Open source software under GPL.
- ▶ Superior (if not just comparable) to commercial alternatives. R has over 5,000 user contributed packages at this time. It's widely used both in academia and industry.
- ▶ Available on all platforms.
- ▶ Large and growing community of peers.
- ▶ Bioconductor: largest (and arguably the best) free collection of software for biological data analysis anywhere.

Why Not Just Use Excel, FlowJo, GraphPad, etc?

1. Reproducibility
   - Its really important that you know what you did.
   - More journals/grants/etc. are also requiring this.
   - The best way to know what you did is to provide all the code.
   - GUI software makes this difficult
   - If you keep a lab notebook, why not do the same thing with your analysis?

2. Flexibility, capabilities and pretty pictures
   - R can handle much larger data sets, much faster, and much more easily than Excel.
   - Huge range of statistical tests, biological data types, etc.
   - Plotting in R is far more sophisticated than any available GUI.
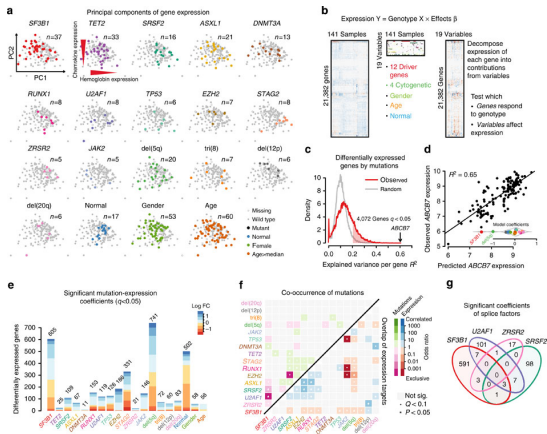
Proof of What I Mean By Pretty Pictures:

Figure 1: Gerstung et al (2015) Nature Communications (CC-BY)

RStudio

Set up a new project

- Click 'file', then 'New project'
- Click 'New directory'm then 'Empy Project', then pick a directory
- With the project set up, click 'file', then 'new' (or ctrl+shift+n)
- Click 'File', 'Save' (ctrl+s)
- Save the file as something meaningful, like `lecture1_examples.R`

Note: for Mac users, where I say 'ctrl', use your weird Mac control key instead.

Quick overview of RStudio

-Figure of interface

Working between the script and console

*Type the following into the console, and press enter:

```
print("Hello")
```

```
## [1] "Hello"
```

- ▶ Now type it into the file window, and with the cursor on that line, press ctrl+enter
- ▶ Messing around in the console is fun.
- ▶ But it's better to keep your work in a file which you save often.

R Basics

Objects

You can assign values to objects:

```
some_number <- 5
some_number + 3
```

```
## [1] 8
```

```
some_number ^ 3
```

```
## [1] 125
```

Take a look in your environment window in RStudio. You can also see what objects are defined using the `ls()` command:

```
ls()
```

```
## [1] "some_number"
```

Basic Data Types

You can find out the type of an object using `typeof()`:

```r
typeof(some_number)
```

```
## [1] "double"
```

```r
some_text <- "To be or not to be"
typeof(some_text)
```

```
## [1] "character"
```

Numeric vs Character

```
some_number + 5
```

```
## [1] 10
```

```
some_text + 5
```

```
## Error in some_text + 5: non-numeric argument to binary operator
```

More Complex Data Types

Data frames, vectors

Installing Packages

CRAN

Bioconductor

Loading in Files

The Iris Data Set

Loading in CSV

Loading in from Excel ?

Basic Statistical Tests

Fisher's T Test

Examining the Results

Basic Plots

Scatter Plot

Box Plots

- Don't use dynamite plots!!

Beeswarm Plots

Other Plotting Packages

Credits

Course Developers

- Kieran O'Neill
- Eva Yap
- Alice Zhu (for next session)

Much material was reused from Software Carpentry's Bootcamp workshops and from Andy Teucher's short R course, both under the terms of the Creative Commons Attribution License.