H3ABioNet

Pan African Bioinformatics Network for H3Africa

16SrRNA Intermediate Bioinformatics Online Course: Int_BT_2019

Introduction to R: best practices for coding

Intro to R: good coding practices

- Start each program with a description of what it does.
- Then load all required packages.
- Improve reproducibility/usability by limiting 'hard-coding'
  - Keep all of the source files for a project in one directory and use relative paths to access them
  - Define all user-specific code and functions at the start of your script

Intro to R: good coding practices

R

```r
input_file <- "data/data.csv"
output_file <- "data/results.csv"

# read input
input_data <- read.csv(input_file)
# get number of samples in data
sample_number <- nrow(input_data)
# generate results
results <- some_other_function(input_file, sample_number)
# write results
```

```r
# check
input_data <- read.csv("data/data.csv")
# get number of samples in data
sample_number <- nrow(input_data)
# generate results
results <- some_other_function("data/data.csv", sample_number)
# write results
write.table("data/results.csv", output_file)
```

Intro to R: good coding practices

- Break code into logical sections with comments to inform script
  - Ctr/Cmd + shift + R
- Create functions for repeated code rather than copy-paste over and over – makes script long and error-prone
- Record sessionInfo()
- Get someone else to review your code
- Use version control, or Github gists!

Good coding practices – useful links

- Good enough practices in scientific computing
- The tidyverse R style guide