**Reading mySQL**

* **mySQL**
  + - used in internet based apps
    - data are structured in
      * databases
      * tables within databases
      * fields within tables
    - each row = a record
* **Example structure**
* **Steps**

1. Install mySQL
2. Install RMySQL

* **Connecting and listing databases**

UcscDb <- dbConnect(MySQL(), user = “genome”, host = “[link]”)

Result <- dbGetQuery(ucscDb, “show databases;”)

dbDisconnect(ucscDb)

UcscDb <- dbConnect(MySQL(), user = “genome”, host = “[link]”)

allTables <- dbListTables(ucscDb)

Length(allTables)

* **Get dimensions of a specific table**

dbListFields(ucscDb, “name of table”)

* **Read from the table**

affyData <- dbReadTable(ucscDb, “[name of table]”)

* **Select a specific subset**
* Remember to close the connection!

**Reading from HDF5**

* **HDF5**
  + - Large datasets
    - Hierarchical data format
    - Groups containing 0 or more datasets and metadata
* **R HDF5 package**

Source(<https://bioconductor.org/biocLite.R>)

biocLite(“rhdf5”)

library(rhdf5)

created = h5createFile(example.h5)

* **Create groups**

Created = h5createGroup(“example.h5”, “foo”)

H5ls(“example.h5”)

* **Write to groups**

A = matrix(1:10, nr = 5, nc = 2)

H5write(A, “example.h5”, “foo/A”)

B = array(seq(0.1, 2.0, by = 1.1), dim = c(5, 2, 2))

Attr(B, “scale”) <- “liter”

H5write(B, “example.h5”, “foo/foobaa/B”)

H5ls(“example.h5”)

* **Write a dataset**
* **Reading data**

H5read(“example.h5”)

* **Writing and reading chunks**

H5write(c(12, 13, 14), “example.h5”, “foo/A”, index = list(1:3, 1))

H5read(“example.h5”, “foo/A”)

**Reading from the web**

* **Webscraping** 
  + - Programmatically extracting data from websites
    - In some cases, this is against the terms of website of the website
* **Reading from the web**

Con = url(“[link]”)

htmlCode = readLines(con)

close(con)

htmlCode

* **Parsing with XML**

Library(XML)

url <- “[link]”

html <- htmlTreeParse(url, useInternalNodes = TRUE)

xpathSApply(html, “//title”, xmlValue)

* **GET from the httr package**

Library(httr)

Html2 = GET)url)

Content2 = content(html2, as = “text”)

parsedHtml = html Parse(content2, asText = TRUE)

xpathSApply(parsedHtml, “//title”, xmlValue)

* **Accessing websites with passwords**

Pg1 = GET(“[link]”, Authenticate(“user”, “password”))

* **Using handles**

Google = handle(<http://google.com>)

Pg1 = GET(handle = google, path = “/”)

Pg2 = GET(handle = google, path = “search”)

**Reading data from APIs**

* Application Programming Interfaces

**Reading from other sources**

* There is a package for everything, just google it
* Interacting more directly with files
  + - File
    - url
    - gzfile
    - bzfile
    - ?connections
    - Remember to close connections
* **Foreign package**
  + - Basic functions
      * Read.arff
      * Read.dta
      * Read.mtp
      * Read.octave
      * Read.spss
      * Read.xport
* Reading images
  + - Jpeg
    - Readbitmap
    - Png
    - EBImage
* Reading GIS data
  + - Rdgal
    - Rgeos
    - Raster
* Reading music data
  + - tuneR
    - seewave