Guía 1: Introducción

Bioinformática - Pasto 2017

Guillermo Torres

Introducción

En el desarrollo de este curso se utilizará lenguaje de programación R para todos nuestros análisis. El curso proporcionará las bases tanto teóricas como prácticas para el aprendizaje básico de R, estadística de datos de alto rendimiento y analisis de RNAseq. Dado que el curso no proporciona un estudio profundo y detallado de la sintaxis de R, se recomienda que el desarrollo de tutoriales para incrementar el conocimiento en programacion basica de R. En esta guía introductoria se proporciona instrucciones paso a paso de como configurar su equipo para comenzar a trabajar en ambiente R de forma amigable.

Instalando R

El primer paso es instalar R. Se puede descargar e instalar R desde el Comprehensive R Archive Network (CRAN). Es relativamente sencillo, pero si necesita mas ayuda la puede encontrar en los siguientes enlaces:

- Instalando R en Windows
- Instalando R en Mac
- Instalando R en Ubuntu

Instalando RStudio

El siguiente paso es instalar RStudio, un programa para visualizar y correr scripts de R. Tecnicamente se pueden correr todo los codigos utilizados en el curso sin instalar RStudio, pero se recomienda el uso de este ambiente de desarrollo integrado (integrated development environment - IDE) por ser más amigable. Las instrucciones para la instalación Aquí! y para Windows, instrucciones extra Aquí!.

Instalando Packages

El primer comando de R que seguro será utilizado es `install.packages`. R solo incluye un set básico de funciones. Debido a la flexibilidad del lenguaje, R permite a los usuarios desarrollar flujos de trabajo específicos los cuales los son organizados en funciones y agrupados como paquetes. Muchas de estas funciones/paquetes estan almacenados en CRAN. Estos paquetes son investigados: chequeados por errores comunes y deben tener una persona que se dedique a mantenerlos. Un paquete puede ser facilmente instalado en R solamente con el nombre. Como ejemplo vamos a instalar el paquete `rafalib` el cual se va a usar para analizar nuestros datos:

```
install.packages("rafalib")
```

Se puede cargar el paquete en la sesion de R usando la función `require` o `library`:

```
library(rafalib)
```

Una vez los paquetes hayan sido instalados, éstos solo seran cargados cada vez que se necesiten.

Aprendiendo lo básico de R

La primera tarea es completar el tutorial `swirl`, con el ánimo de familiarizarse con la sintaxis básica del lenguaje R, dado que en el curso estos temasno serse cubiertos en detalle y se enfocará mas en temas relacionados con estructuras de datos como listas, data frames, vectores numericos, loops-for, como crear funciones y como usar las funciones `apply` y `replicate`, entre otros.

El Tutorial swirl enseña a programar en R de manera interactiva directamente sobre la consola de R. Una vez R haya sido instalado, se puede instalar `swirl` y correrlo de la siguiente manera:

```
install.packages("swirl")
library(swirl)
swirl()
```

No obstante, otra alternativa puede ser tomar la clase interactiva de la Code School: Try R. Adicionalmente existen muchos recursos gratis y guías de referencia en internet; por ejemplo:

- Quick-R: Un recurso online que permite ejercitarce en manejo de datos, estaditica básica y visualización
- R reference card PDF by Tom Short

Dos cosas claves que se necesitan saber aserca de R es que uno puede obtener información sobre una función utilizando `help` o `?`, de la siguiente manera:

```
?install.packages
help("install.packages")
```

y los caracteres numeral o `hash` representan comentarios, por lo que el texto que sigue al caracter no será interpretado como codigo R:

```
## Este es un comentario... Comentarios aqui...
```

Importando datos a R

El primer paso en el analisis de datos es leer los datos en R. Existen varias formas de hacerlo y aquí discutiremos 3 de ellas, sin embargo solo necesitan aprender una de ellas ;).

En ciencias, pequeños set de datos tipicamente pueden ser almacenados y analizados en Excel. Aunque existen paquetes de R que permiten leer archivos Excel (.xls), generalmente se recomienda uno evita este tipo de datos y almacena la información en archivos más ligeros como quellos delimitados por comas (Comma-Separated Value/csv) o delimitados por tabuladores (Tab-Separated Value/tsv/txt). Estos formatos llamados de texto-plano son fáciles de manipular y para compartir con colaboradores dado que no se requiere de ningún software comercial para poder visualizarlos o trabajar sobre ellos.

Para los siguientes ejercicios trabajaremos con un data set simple: `female mause weights` pero el primer paso es definir la ubicación del archivo, es decir conocer su ruta o `*path*`.

Paths y directorios de trabajo

Cuando se trabaja en R es útili definir y/o conocer el directorio de trabajo *working directory*. Este es el directorio o folder en el cual R guardará y buscarán los archivos por defecto. Uno puede saber cual es el directorio de trabajo tipeando:

```
getwd()
```

Se puede cambiar el directrio de trabajo usando la función `setwd`, o cambiando a través de RStudio, dando click sobre la pestaña `Session`.

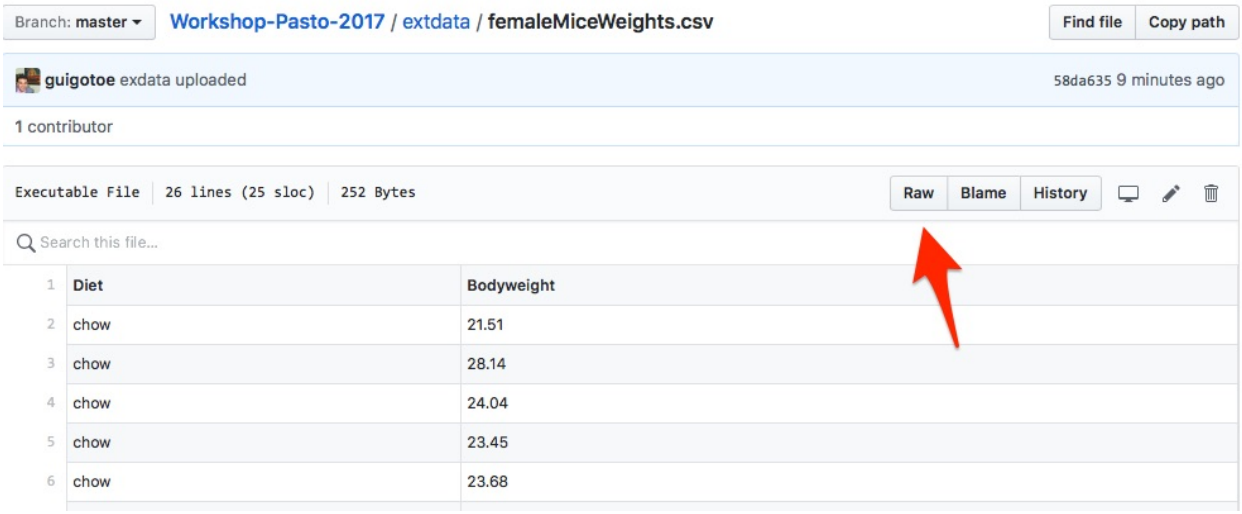

Figure 1: GitHub page screenshot.

```
setwd("/Users/guillermotorres/Documents/Proyectos/2017/12PastoWorkshop/CourseLab/ws")
```

Las funciones que leen y escriben archivos (varias en R) asumen que los archivos estan o seran grabados en el directorio de trabajo. Como recomendación para principiantes es importante que se defina el directorio de trabajo y que ahí se guarden los datos que se vayan a utilizar. No obstante, uno puede proporcionar a R la ruta completa (absoluta) del archivo independientemente de la ruta del directorio de trabajo.

Projectos en RStudio

Una de las formas más fáciles para la organizacion de nuestro trabajo es comenzar un proyecto en RStudio (click en "File" y luego en "New Project"). Cuando un nuevo proyecto se crea, es necesario seleccionar un folder asociado al proyecto el cual se convertira en el directorio de trabajo. Todos los datos deberán ser descargados y almacenados en este folder.

Opción 1: Leer archivos desde internet

Se puede navegar el archivo `femaleMiceWeights.csv` visitando el el directorio de extdata de la siguiente direccion en internet: GitHub: https://github.com/guigotoe/Workshop-Pasto-2017/tree/master/extdata/. Para navegar el archivo se necesita hacer click sobre *Raw* en la esquina superior derecha de la página y luego usar la función de "guardar como" del navegador para guardar el archivo en su computador. Asegurese de que el archivo guardado sea en formato CSV.

Opción 2: Descargar los archivos a el directorio de trabajo

Multiples razon existen para trabajar con una copia local de los datos en vez de hacerlo conectado a internet. Muchas veces trabajar localmente facilita la ejecución de ensayos de reproducibilidad. El archivo se puede descargar de internet via navegador hacia un folder local. Una vez guardado el archivo en el directorio de trabajo este se puede leer de la siguiente manera:

```
dat <- read.csv("femaleMiceWeights.csv")
```

Si no aparece ningun mensaje, entonces significa que probablemente el archivo fue leido satisfactoriamente.

Opción 3: Descargar el archivo desde R.

Muchos de los archivos de trabajo estan almacenados en GitHub: https://github.com/guigotoe/Workshop-Pasto-2017/tree/master/extdata/. Se pueden guardar estos archivos directamente desde internet al computador usando R. En este ejemplo usaremos la función `download.file` del paquete `downloader` para descargar el archivo a una localización específica de nuestro computador y luego leerlo. Es posible asignarle un nombre aleatorio y un directorio aleatorio utilizando la función `tempfile` o asignarle un nombre específico.

```
library(downloader) ## Usar install.packages para instalar el paquete downloader
dir <- "https://github.com/guigotoe/Workshop-Pasto-2017/tree/master/extdata/"
filename <- "femaleMiceWeights.csv"
url <- paste0(dir, filename)
if (!file.exists(filename)) download(url, destfile=filename)
```

El archivo se guardará en el directorio de trabajo. Luego podemos proceder como en la opcion 2

```
dat <- read.csv(filename)
```

Opción 4: Descargar datos de un paquete (Avanzado)

Muchos datasets que se utilizarán en este curso estan disponibles como paquetes "custom-built" en GitHub. La razón por la que usamos GitHub, en vez de CRAN, es por que en este repositorio no es necesario tanta rigurosidad en la estructura del paquete como en CRAN; por lo tanto GitHub brinda una mayor flexibilidad.

Para instalar paquetes de GitHub se necesita tener instalado el paquete `devtools`:

```
install.packages("devtools")
```

Para los usuarios de Windows!!: Para usar `devtools` se necesita tener instalado `Rtools`. En general se necesita instalar los paquetes como administrador. Una manera de hacerlo es comenzar R como administrador. Si no se tienen los permisos para esto entonces es un poco más complicado. Más información aquí!.

```
library(devtools)
install_github("genomicsclass/dagdata")
```

El archivo con el cual queremos trabajar está incluido en el paquete que acabamos de instalar. Una vez instalado el paquete el archivo esta en el computador, sin embargo encontrarlo requiere conocimientos más avanzados. A continuación la solución:

```
dir <- system.file(package="dagdata") # Extrae la localización del paquete
list.files(dir)
list.files(file.path(dir,"extdata")) # la carpeta extdata esta en este directorio
```

Y ahora estamos listos para leer el archivo:

```
filename <- file.path(dir,"extdata/femaleMiceWeights.csv")
dat <- read.csv(filename)
```