Introducción TPs Bionformática

Licán Martinez & Ezequiel Perez

BCyM 2022

# 1. Conceptos básicos en bioinformática

## 1.1 Los archivos FASTA

## 1.2 El NCBI

## 1.3 BLAST Y GEO

# 2. El lenguage de programación R

Explicar resumídamente la historia del lenguage R, sus ventajas, desventajas y controversias.

## 2.1 Instalación de R y de Rstudio

Todos los análisis que se presentan a continuación requieren el uso del entorno R. Es muy probable que ya tenga R instalado en su computadora, pero si no fuera así, o si es necesario actualizar la versión (recomendable). Para instalar o actualizar R, ir a <https://cran.r-project.org/> y seleccionar el paquete base para su sistema operativo.

RStudio es un entorno que facilita las tareas de scripting y ejecución de R. Para instalar o actualizar R, ir a <https://rstudio.com/products/rstudio/download/> y seleccionar RStudio Desktop, Open Source License.

## 2.2 Sintáxis R

Aprender un lenguaje de programación es como aprender un segundo idioma (sólo que más simple). Si visitamos un pais extranjero , podríamos aprender frases para salir del paso sin entender como está estructurado el lenguaje. De forma similar, si sólo queremos hacer un par de cosas con R (p. ej. dibujar gráficos), probablemente sea suficiente con estudiar a partir de unos cuantos ejemplos.

Sin embargo, si lo que queremos es aprender realmente bien un nuevo lenguaje, tendremos que aprender su sintaxis y gramática: conjugación verbos, estructuras gramaticales, etc. Lo mismo es cierto para R: si queremos aprender a programar en R, tendremos que aprender la sintaxis y gramática.

### 2.2.1 Expresiones y asignación

El código R está compuesto por una serie de expresiones. A continuación se muestran algunos ejemplos de cada una de estas expresiones.

* Expresiones aritméticas.

2 + 1 # suma

## [1] 3

2\*6 # multiplicación

## [1] 12

2/4 # división

## [1] 0.5

3^3 # potencia

## [1] 27

120%%5 # resto de una división

## [1] 0

* Expresiones de asignación.

Podemos crear nuevos “objetos” con el operador de asignación “<-” o también con “=”. Todas las instrucciones de asignación dónde creamos objetos, tienen la misma forma:

nombre\_objeto <- valor

Orwell <- 1984  
Huxley = "A Happy World"  
  
# recuperando el valor de los objetos:  
Orwell

## [1] 1984

Huxley

## [1] "A Happy World"

* Expresiones lógicas.

2 == 2 # en este caso usamos el "==" como símbolo de equivalencia, dado que el "=" esta apartado para la expresión de asignación.

## [1] TRUE

5 > 6

## [1] FALSE

BCyM <- "grandioso"  
BCyM == "aburrido"

## [1] FALSE

* Expresiones de control (condicionales e iterativas)

La expresión condicional evalua la veracidad de la expresión lógica entre paréntesis “if (EXP. LÓGICA)” y luego ejecuta el código del cuerpo “if” si “EXP. LÓGICA = TRUE” o del cuerpo “else” si "“EXP. LÓGICA = FALSE”

# Expresión condicional  
if (1 > 2) {"mayor"} else {"menor"}

## [1] "menor"

La expresión iterativa “while” -itera- un bloque de código hasta que cierta expresión condicional se vuelve FALSE. La expresión iterativa “for” -itera- un bloque de código hasta que el objeto “i” haya pasado por todos los valores de un vector.

# Expresión iterativa while.  
i = 0  
while (i<4){  
 print(i)  
 i = i+1  
}

## [1] 0  
## [1] 1  
## [1] 2  
## [1] 3

# Expresión iterativa for.  
for (i in 1:4){  
 print(i)  
}

## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4

### 2.2.2 Objetos

### 2.2.3 Tipos de datos

## 2.4 Estructura de datos

## 2.5 Estructuras de control

## 2.6 Materiales adicionales