Class06 - R Functions Lab

Gabriella Tanoto (A18024184)

Table of contents

Experimenting with R Functions

1. Basic FUNction

Let's make an `add` function!

An R function always has 3 things:

- name (pick!)
- input arguments (can be as many as we want, separated by ',' comma)
- the body (R codes that do the work, inside '{}')

```
add <- function(x, y=1) {    #y=1 is a "default"
  x + y      #this is the BODY!
}
```

Try out the function!

```
add(1, 3)
```

```
[1] 4
```

```
add(5, c(1,2,3,4,5))
```

```
[1]  6  7  8  9 10
```

`add(, c(2,3,4))` -> will fail, cuz X is a required argument.

Functions can have a "required" input argument and "optional" if it has got a default. Defaults have an '=' sign.

`add(1,2,5)` #will fail, cuz we only made the function add 2 numbers (x and y)

2. Generating DNA!

Q. Write a function to return a DNA seq of a user precified length. Cal it: `generate_dna()`

We can use the `sample()` function to help.

```
# something like: generate_dna <- function(size=5) {}

## Now, I try `sample` fnction.
DNA <- c("A", "C", "T", "G")
sample(DNA, 10, replace=TRUE) # It did work! Yay! so now make the fnctn!
```

```
 [1] "T" "G" "C" "A" "T" "T" "A" "T" "G" "C"
```

Now that we know the building blocks, make it into a function!

```
generate_dna <- function(x=20) {
  DNA <- c("A", "C", "T", "G")
  seq <- sample(DNA, size=x, replace=TRUE)
}

generate_dna() #use default.
generate_dna(111) #use inputted value!
```

Let's polish it, so it doesn't generate with "Quote" marks. I want it to be like "ACCTG-GAATGGC" so it can be BLASTed!!!

```
generate_DNA <- function(x=20, blast=TRUE) {
  DNA <- c("A", "C", "T", "G")
  seq <- sample(DNA, size=x, replace=TRUE)
  if(blast){
    seq <- paste(seq, collapse="")
  }
  return(seq)
}

#Trying it outtt:
generate_DNA()
```

```
[1] "TTTTCTTCAACCACTTGACA"
```

```
generate_DNA(blast = FALSE)
```

```
 [1] "A" "C" "C" "A" "C" "G" "G" "T" "A" "T" "C" "G" "A" "T" "A" "A" "A" "A" "C"
[20] "A"
```

NOTES TO SELF:

- You HAVE to specify the <seq> at both the `if` and the non-`if`.
- Then, you have to include the `return` at the end, so the function works both ways —with and w/out if.

3. Generate Protein!

We can get the list of the 20 natural Amino Acids from **bio3d** <- gotta install (`install.packages`), then open (`library`)!

```
AA <- bio3d::aa.table$aa1[1:20]
AA #check if u got it!
```

```
 [1] "A" "R" "N" "D" "C" "Q" "E" "G" "H" "I" "L" "K" "M" "F" "P" "S" "T" "W" "Y"
[20] "V"
```

Q. Write the protein-sequence-generating function that returns a sequence of user's generated length!

```
generate_protein <- function(x=50, blastp=TRUE) {
  AA <- bio3d::aa.table$aa1[1:20] #the AA's are vectors.
  prot.seq <- sample(AA, size=x, replace=TRUE)

  if(blastp){
    prot.seq <- paste(prot.seq, collapse="")
  }
  return(prot.seq)
}

#Try out:
generate_protein(33)
```

```
[1] "VVLMDENEVMNHSFYFANKTMHNQFFIRQAQVS"
```

```
generate_protein(blastp=FALSE)
```

```
 [1] "I" "A" "Q" "M" "W" "L" "C" "R" "A" "P" "A" "W" "D" "W" "K" "P" "H" "E" "S"
[20] "Q" "F" "C" "Q" "D" "I" "N" "C" "E" "A" "L" "M" "N" "M" "G" "M" "W" "M" "T"
[39] "H" "G" "Y" "V" "K" "K" "R" "Y" "C" "T" "A" "L"
```

Q. Generate protein with multiple random sequences of *6 to 12*!

We want R to produce multiple sequences. Do this by either: - *editing* and *adding* the function body code (for a loop), OR - use the R **apply** family of utility fctn.

```
#sapply(X, FUN) <- X is VECTOR. FUN is function. Stands for Simple Apply
prot6to12 <- sapply(6:12, generate_protein)
```

We have to make it into a FASTA format in order to BLAST it.

```
id.line <-paste(">ID.", 6:12,"\n", prot6to12, sep="")
id.line
```

```
[1] ">ID.6\nKIKPEW"        ">ID.7\nKSELLHC"       ">ID.8\nREFHKRIQ"
[4] ">ID.9\nWPMMNQPYW"     ">ID.10\nVQCTGVYAWP"   ">ID.11\nASDADKTELAE"
[7] ">ID.12\nFRVKMPVAKPWL"
```

```
#Concatenating the "id.line" makes the "\n" into an [ENTER]

cat(id.line, sep="\n")
```

```
>ID.6
KIKPEW
>ID.7
KSELLHC
>ID.8
REFHKRIQ
>ID.9
WPMMNQPYW
>ID.10
VQCTGVYAWP
>ID.11
ASDADKTELAE
>ID.12
FRVKMPVAKPWL
```

Q. Determine if we can find the sequences in nature!

I BLASTp searched my FASTA format protein. Only ones that are not 100% Coverage and 100% Identity.

- Non-unique: 6, 7
- Unique: 8, 9, 10, 11, 12

My random sequences that were 8, but others are generally 9 or above. There are very **LOW** chance of a *random* amino acid sequence that are higher than 9 that is exactly the same in nature.

** MHC presents 9 amino acid sequences too. ** BLASTp has a window of 9 – if they are similar for 9 or above, then it's probably a related protein!