Class06: Writing functions in R

Jihyun In (PID: A16955363

Introduction

R code

```
add <- function(x, y=10, z=0){
  x + y + z
}
```

I can just use this funciton

```
add (1,100)
```

```
[1] 101
```

```
add(x=c(1,2,3,4), y=100)
```

```
[1] 101 102 103 104
```

Functions can have "required" input arguments and "optional" input arguments. The optional arguments are defined with an equals default value (`y=0`) in the function definition.

```
add(x=1, y=100, z=10)
```

```
[1] 111
```

Generate DNA sequence

> Q. Write a function to return a DNA sequence of a user specifed length. Call it `generate_dna()`

The `sample()` can help here.

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

students <- c("jeff", "jeremy", "peter")

sample(students, size = 5, replace=TRUE)
```

```
[1] "jeremy" "jeff"   "peter"  "jeff"   "peter"
```

Now work with `bases` rather than `students`

```
bases <- c("A", "C", "G", "T")
sample(bases, size=10, replace=TRUE)
```

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

Now I have a working 'snippet' of code, so I can use this as the body of my first function version here:

```
generate_dna <- function(size=5){
  bases <- c("A", "C", "G", "T")
  sample(bases, size=size, replace=TRUE)
}
```

```
generate_dna(size=100)
```

```
  [1] "C" "T" "T" "T" "C" "A" "G" "C" "A" "T" "C" "G" "T" "C" "G" "T" "G" "G"
 [19] "C" "G" "G" "T" "A" "G" "C" "G" "C" "T" "C" "C" "C" "G" "T" "C" "C" "G"
 [37] "C" "G" "T" "A" "T" "T" "T" "A" "C" "A" "A" "G" "G" "C" "G" "G" "G" "A"
 [55] "C" "C" "A" "A" "G" "G" "A" "A" "A" "T" "A" "C" "T" "A" "A" "G" "G" "G"
 [73] "T" "C" "G" "G" "G" "G" "T" "G" "C" "C" "T" "T" "T" "A" "C" "T" "G" "C"
 [91] "C" "G" "A" "C" "C" "G" "A" "C" "G" "A"
```

```
generate_dna()
```

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

I want the ability to return a sequence like "AGTACCTG" string, i.e. a one element vector where the bases are all together.

```
generate_dna <- function(size=5, together=TRUE){
  bases <- c("A", "C", "G", "T")
  sequence <- sample(bases, size=size, replace=TRUE)
  if (together){
    sequence <- paste(sequence, collapse = "")
  }
  return(sequence)

}
```

```
generate_dna(together=FALSE)
```

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

3. Generate Protein function

we can get the set of 20 natural amino-acids from the **bio3d** package

q. Write a protein sequence generating function that will return sequences of a user-specified length?

```
generate_protein <- function(size=5, together=TRUE){
 aa <- bio3d::aa.table$aa1[1:20]
 sequence <- sample(aa, size=size, replace=TRUE)

  ## Optionally return a single string
  if (together){
    sequence <- paste(sequence, collapse = "")
  }
  return(sequence)
}
```

Q. Generate random protein sequences of length 6 to 12 amino acids.

```
#errored code
#generate_protein(size=6:12)
```

We can fix this inability to generate multiple sequences by either editing and adding to the function body code (eg. for a for loop) or by using the R **apply** family of utility functions

```
ans <- sapply(6:12, generate_protein)
ans
```

```
[1] "HNQGAE"       "QMAHYCH"      "HSDMDTGM"     "QEIVAAVGL"    "CGEEKMAPPF"
[6] "NDCPQHHSSLL"  "NTGQRIDDTFKC"
```

It would be cool and useful if I could get FASTA format output.

```
cat(ans, sep="\n")
```

```
HNQGAE
QMAHYCH
HSDMDTGM
QEIVAAVGL
CGEEKMAPPF
NDCPQHHSSLL
NTGQRIDDTFKC
```

I want this to look like

```
>ID.6
CNSTRV
>ID.7
NGCVYMV
>ID.8
QATSNMQI
```

```
with.id <- paste(">ID.",6:12,"\n", ans, sep="")
cat(with.id, sep="\n")
```

```
>ID.6
HNQGAE
>ID.7
QMAHYCH
>ID.8
HSDMDTGM
>ID.9
QEIVAAVGL
>ID.10
CGEEKMAPPF
>ID.11
NDCPQHHSSLL
>ID.12
NTGQRIDDTFKC
```

> Q. Determine if these sequences can be found in nature or are they unique? Why or why not?

I BLASTp searched by FAFST format sequences against the sequences with length 6, 7, 8, were not unique with 100% coverage and 100% identity.

sequences with length 9,10,11,12 are unique and can't be found in the databases.