Lab 6 - R Functions

Jessica Le (PID: A17321021)

Today we will get more exposure to functions in R. We call functions to do all our work and today we will learn how to write our own.

A first silly function

Note that arguments 2 and 3 have default values (because we set y=0 and z=0) so we don't have to supply them when we call our function.

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

Can I use this function?

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

```
[1] 2
```

```
add(1, c(10,100))
```

```
[1]  11 101
```

```
add(100)
```

```
[1] 100
```

```
add(100, 10, 1)
```

```
[1] 111
```

A second more fun function.

Let's write a function that generates random nucleotide sequences.

We can make use of the in-built `sample()` function in R to help us.

```
sample(x=1:10, size=11, replace=TRUE)
```

```
 [1]  8  5  4  8  4  1  5  7  5  2 10
```

> Q. Can you use 'sample()' to generate a random nucleotide sequence of length 5.

```
sample(x=c("A", "G", "C", "T"), size=5, replace=TRUE)
```

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

> Q. Write a function `generate_dna()` that makes a nucleotide sequence of a user specified length.

Every function in R has at least 3 things:

- a **name** (in our case 'generate_dna()')
- one or more **input arguments** (the length of sequence we want)
- a **body** (R code that does the work)

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

```
generate_dna(20)
```

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

> Q. Can you write a `generate_protein()` function that returns amino acid sequence of a user requested length? generate_protein

Install bio3d package using `install.packages()` to access the amino acid sequences. - bio3d::aa.table - bio3d::aa.table$aa1

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

```
generate_protein(10)
```

```
 [1] "P" "P" "I" "Q" "P" "R" "I" "S" "I" "Y"
```

I want my output of this function not to be a vector with one amino acid per element but rather a single string.

```
bases <- c("A", "G", "C", "T")
paste (bases, collapse="----")
```

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

```
bases <- c("A", "G", "C", "T")
paste (bases, collapse="")
```

```
[1] "AGCT"
```

Trying to collapse the `generate_protein` function.

```
generate_protein <- function(length=5) {
  aa <- bio3d::aa.table$aa1[1:20]
  s <- sample(aa, size=length, replace=TRUE)
  paste(s, collapse="")
}
```

```
generate_protein()
```

```
[1] "VLIFD"
```

Q. Generate protein sequences from length 6 to 12?

```
generate_protein(length=6)
```

```
[1] "HEWHNT"
```

```
generate_protein(length=7)
```

```
[1] "PTHKYSC"
```

```
generate_protein(length=8)
```

```
[1] "ECVDKTTS"
```

We can use the useful utility function `sapply()` to help us "apply" our function over all the values 6 to 12.

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

```
[1] "APTYMM"       "CSVHKNI"      "QCIVGVPA"     "FSKIMDEAF"    "LKPEMGNVFT"
[6] "SADIYLQPGCP"  "TTPHLPLQASEC"
```

```
paste(">ID", 6:12)
```

```
[1] ">ID 6"  ">ID 7"  ">ID 8"  ">ID 9"  ">ID 10" ">ID 11" ">ID 12"
```

To remove the space between ID and the value.

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

```
>ID.6
APTYMM
 >ID.7
CSVHKNI
 >ID.8
QCIVGVPA
 >ID.9
FSKIMDEAF
```

```
 >ID.10
LKPEMGNVFT
 >ID.11
SADIYLQPGCP
 >ID.12
TTPHLPLQASEC
```

> Q. Are any of these sequences unique in nature - i.e. never found in nature? We can search "refseq-protein" and look for 100% Ide and 100% coverage matches with BLASTp.

All the sequences are unique in nature because no significant similarity were found using BLASTp.