R Functions

Bayah Essayem (A17303992)

Today we'll 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 sillly function

Note that argument 2 and 3 have default values (because we set y=0 and z=0)

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

You can't use the function until you press run, if you don't run it R won't know the function you just created. Otherwise doing [add(1,1)] will give an error.

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

```
[1] 2
```

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

```
[1]  11 101
```

Without a y argument, you can't use this function. It'll give an error because the function has been defined to have an x and y. ex: add(100)

```
##add(100)
```

But, if you give it a y it'll work:

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

Now, let's try:

```r
add(100)
```

```
[1] 100
```

If you do this:

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

You get an error because the function has been defined to only have 2 variables (x and y) so even if you try to add an additional one it won't work, until you define a z:

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

Now, this can work:

```r
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 here.

```r
sample(x=1:10, size=9, replace = T)
```

```
[1] 8 1 6 2 1 4 6 4 4
```

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

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

```
[1] "T" "C" "T" "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) a **name** (in this case 'generates_dna()') b) one or more **input arguments** (the "length" of the sequence we want) c) a **body** (R code that does the work)

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

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

```
aa <- bio3d::aa.table$aa1[1:20]
```

```
generate_protein <- function (length=5) {
  amino_acids <- (aa)
  sample (amino_acids, size = length, replace = T)
}
```

```
generate_protein (10)
```

```
 [1] "H" "Y" "E" "K" "N" "T" "D" "V" "E" "G"
```

```
generate_protein (100)
```

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

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

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

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

```
generate_protein <- function (length=5) {
  amino_acids <- (aa)
s <- sample (amino_acids, size = length, replace = T)
paste(s, collapse="")
}
```

```
generate_protein()
```

```
[1] "MNSQR"
```

Q. Generate protein sequences from length 6 to 12?

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

```
[1] "QFTSYS"
```

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

```
[1] "DRLKIPE"
```

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

```
[1] "GQERQRAD"
```

We can use the useful utility function 'sapply()' to help us "apply" our function over all the values 6 to 8

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

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

```
>ID.6
GGLGFM
 >ID.7
SEHTIHW
 >ID.8
PKAMRQAE
 >ID.9
VYCHQDHVH
 >ID.10
LWYGMHTYMD
 >ID.11
ASELVAILCHH
 >ID.12
PMKLAMWYQSAN
```

> Q. Are only one 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

After searching in BLASTp, I found that many of my generated sequences are found in nature, not just one. This includes: uncharacterized protein LOC119732930 [Patiria miniata], uncharacterized protein LOC119723897 [Patiria miniata], Calx-beta domain-containing protein [Gimesia fumaroli].