R functions

Aaron (PID:A17544470)

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

A first test function

Note that argument 2 and 3 have default values as we set y = 0 and z = 0 so that it is not necessary to have them in our function

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

Does this work?

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

```
[1] 2
```

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

```
[1]  11 101
```

If the function add is not found, you need to run the function so add does something.

```
add(100)
```

```
[1] 100
```

Argument "y" is missing, without a default value

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

```
[1] 111
```

A second test function

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

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

sample()

```
sample(x = 1:10, size = 1)
```

```
[1] 10
```

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

```
 [1]  8 10  3 10  4  4  7  6  6  8  3
```

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

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

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

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

```
generate_dna <- function(length = 5){

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

}

generate_dna(10)
```

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

Prof method

```
generate_dna2 <- function(length2=5){
  bases <- c("G","T","C","A")
  sample(bases, size=length2, replace = T)

}

generate_dna2(10)
```

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

Every function in R has at least 3 things:

- a **name** (in our question "generate_dna")
- **input arguments** (the "length" of the sequence we want)
- a **body** that does the work (R code)

  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(length3 = 5) {
  sample(aa, size=length3, replace = T)
}

generate_protein(10)
```

```
 [1] "A" "P" "F" "I" "N" "A" "H" "G" "M" "P"
```

I want the output of this function to be one string instead of a vector with an amino acid per element.

```
paste(generate_protein(), collapse = "")
```

```
[1] "RREEL"
```

Combining leads to:

```
generate_protein <- function(length3 = 5) {
  aa <-bio3d::aa.table$aa1[1:20]
  s <- sample(aa, size=length3, replace = T)
  paste(s, collapse = "")
}
```

```
generate_protein()
```

```
[1] "YACDF"
```

Q. Generate protein sequences from length 6 to 12?

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] "EPMRWP"       "NLFLNDD"      "SWRCRYAA"     "AHRNVIDDN"    "DTCDALAVYG"
[6] "LFFQGWLGNEP"  "CEKAPWHDRFIY"
```

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

```
>ID6
EPMRWP
>ID7
NLFLNDD
>ID8
SWRCRYAA
>ID9
AHRNVIDDN
>ID10
DTCDALAVYG
>ID11
LFFQGWLGNEP
>ID12
CEKAPWHDRFIY
```

Q. Are any of these sequences unique in nature - i.e. never found in nature. We can search "refseq-protein" and look for 100% identical and 100% coverage.

The lower numbers have matches but the higher numbers do not. The reasoning is that there are just so many combinations at these lengths that it is virtually impossible to match them.