

Class 6: R Functions

Josefina O'Toole (PID: A16978557)

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.

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

Can I just use this? Only after pressing play on the function above to create the function.

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

```
[1] 2
```

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

```
[1] 11 101
```

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

```
[1] 11 101
```

A first silly function

Let's give our function a default.

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  
}  
  
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 here.

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

```
[1] 9 1 2 7 8 3 10 3 4 8 9
```

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

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

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

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** (that does the work)

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

generate_dna(7)
```

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

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

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

generate_protein(7)
```

```
[1] "C" "L" "I" "D" "Q" "W" "E"
```

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.

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

generate_protein(7)
```

```
[1] "EGTIFGG"
```

Q. Generate protein sequences from length 6 to 12?

```
generate_protein(6)
```

```
[1] "SNFGSD"
```

```
generate_protein(7)
```

```
[1] "EPIAQHN"
```

```
generate_protein(8)
```

```
[1] "MELCPWVA"
```

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)
```

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

```
>ID.6
MGMFPV
>ID.7
VMANGPD
>ID.8
LSRSPDRF
>ID.9
FFFLMGGGT
>ID.10
DCAHDMEAAE
>ID.11
NYVHREFYIYH
>ID.12
WIFMQIEHWEKH
```

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.

It could be found in nature because the results showed 100% identity and 100% coverage.