# R functions

Joseph Elmaghraby (PID: A16788229)

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.

## Afirst silly function

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

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

Can I just use this

```
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 start a function that generate nucleotide sequences.

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

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

```
[1]  9  6  2  1 10  7  3  8  5
```

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

```
 [1]  4 10 10  6  9  3  2  6  9  1  3
```

> Q. Can you use 'sample()' to generate a nucle0tide sequence of length 5?

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

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

> 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 the sequnce we want)

-a **body** ( R code that does work )

```r
generate_DNA <- function(length=5) {

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

}
```

```r
generate_DNA(38)
```

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

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

Can you write a ‘generate_protein()’

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

}

generate_protein(45)
```

```
[1] "GWKGYYFWQPYWCPTQRSAANFFYLQDMEMYFYHEYPEIGFYNNC"
```

```r
generate_protein(23)
```

```
[1] "DWSFGSFDHYTSTICQQDFVQHQ"
```

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

```r
bases<- c(aa)
paste(bases, collapse="")
```

```
[1] "ARNDCQEGHILKMFPSTWYV"
```

Q. Generate protein sequnces from length 6 to 12?

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

```
[1] "MEQEWM"
```

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

```
[1] "VHIQSQW"
```

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

```
[1] "LSYKVWGV"
```

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] "SPGGMF"       "NFKSNLR"      "KDMGMLIG"     "FWKMYVWRV"    "SKVWGGYGYI"
[6] "AENPQIECSGF"  "YEAVSTFTCLLD"
```

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

```
>ID.6
SPGGMF
 >ID.7
NFKSNLR
 >ID.8
KDMGMLIG
 >ID.9
FWKMYVWRV
 >ID.10
SKVWGGYGYI
 >ID.11
AENPQIECSGF
 >ID.12
YEAVSTFTCLLD
```

> Q. Are any of these sequences unique in nature - i.e. never found in nature. We can search "refseq-Protein" and look 100% Id and 100% coverage matches with Blastp? > Nosignificant result was found.