BIMM143 Class 06

Yaniv Iny (PID:A18090586)

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 just use this? You can only use it once you already run the function above so that it goes into the R brain.

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

```
[1] 2
```

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

```
[1]  11 101
```

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

```
[1] 111
```

A second more fun function

Lets write a function that generates random nucleotide sequences.

We can make use of the inbuilt sample function in R to help us here.

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

```
[1]  4  5  3  7  1 10  8  2  6
```

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

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

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

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

```
[1] "A" "C" "T" "A" "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 sequence we want)
- a **body** (that does the work)

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

```
generate_dna(10)
```

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

```
generate_dna(100)
```

```
  [1] "A" "G" "G" "C" "C" "A" "T" "G" "T" "A" "G" "T" "C" "A" "C" "G" "G" "A"
 [19] "T" "A" "T" "G" "G" "A" "T" "C" "G" "T" "C" "T" "T" "C" "A" "A" "T" "T"
 [37] "T" "A" "A" "C" "G" "A" "T" "A" "C" "G" "T" "T" "G" "C" "T" "G" "T" "C"
 [55] "G" "A" "A" "A" "G" "C" "T" "A" "G" "C" "G" "G" "G" "T" "C" "T" "C" "A"
 [73] "G" "T" "G" "G" "G" "T" "T" "C" "C" "T" "A" "C" "A" "C" "G" "G" "G" "A"
 [91] "C" "G" "G" "A" "C" "A" "A" "A" "G" "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){
  aa<-bio3d::aa.table$aa1[1:20]
  s <- sample(aa, size=length, replace=T)
}
```

```
generate_protein(10)
```

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

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

```
[1] "ACGT"
```

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

Q. Generate protein sequences from length 6-12?

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

```
[1] "YAHYVD"
```

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

```
[1] "SMDNTRI"
```

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

```
[1] "FKIEEEYR"
```

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] "NNKTRA"       "QAAIHQD"      "GRLMMNFY"     "PNEHCKVQM"    "GTTWRAEFNQ"
[6] "HDMFTYTVHTY"  "HHSYMSIETANF"
```

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

```
>ID.6
NNKTRA
>ID.7
QAAIHQD
>ID.8
GRLMMNFY
>ID.9
PNEHCKVQM
>ID.10
GTTWRAEFNQ
>ID.11
HDMFTYTVHTY
>ID.12
HHSYMSIETANF
```

> 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 mathes with BLASTp

With ID.6,Id.7, and Id.8 we see that there is multiple different matches that are both 100% Identity and 100% coverage as the sequences get longer we start to see more unique sequence. As we move to sequence 8 through 12 there is no matches that come up with 100% identity and coverage meaning that these sequences are unique in nature due to the high amount of differnet possibilities that can occur.