R Functions

Thomas Bailey

today we will get more exposure to functions in R. We call functions to do all our work in R.

##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 functions.

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

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

```
[1] 2
```

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

```
[1]  11 101
```

Can I just use this

```
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=9)
```

```
[1] 9 3 5 4 2 1 8 6 7
```

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

```
 [1] 7 2 2 1 3 9 3 9 4 5 4
```

> Q. Can you use `sample()` to generate a random nucleotide sequence of length 5. ##yes you can set x = to the bases the set size to 5 and put true on replace

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

```
[1] "T" "G" "G" "A" "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 **name** (in our case "generate_dna") -one ore more **input arguments** (the "length" of sequence we want) -a **body** (that does the work)

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

}
```

```
    generate_dna(17)
```

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

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

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

```r
bases<- c("A", "G", "C", "T")

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

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

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

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

We can use the

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

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

```
>ID6
K V F I G N
 >ID7
F S L H Q V V
 >ID8
V N D Y N T A H
 >ID9
W V P Y M W L G H
 >ID10
V C K G N T M K Y F
 >ID11
Q F L M A N W G K P Y
 >ID12
L T C N K R D Y D Q T V
```

> 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.

In looking up in blastp we found many similar sequences but with high E-Values suggesting that even though they might say they have 100% Ide and 100% Coverage that they are still not naturally found out in nature so they are unique sequences.