R functions

Nhi To (PID: A18053310)

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

```
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 write a function that generates random nucleotide sequences

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

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

```
[1]  2  5  3  9  4  6 10  7  1
```

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

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

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

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

```
[1] "A" "T" "A" "G" "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** (R code that does the work)

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

```r
generate_dna(10)
```

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

> Can you write a 'generate_protein() function that returns amino acid sequence of a use requested length?

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

```
generate_protein(15)
```

```
 [1] "R" "N" "F" "E" "R" "A" "Q" "I" "I" "C" "A" "I" "I" "Q" "S"
```

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

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

```
[1] "AGCG"
```

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

```
generate_protein()
```

```
[1] "GDARW"
```

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] "CIEEVL"       "KGDYRSK"      "GPEELRLM"     "IKVSQQEEK"    "KWPCDQGCWS"  
[6] "WNQPLIVNWVW"  "YQYWYSHTCCEL"
```

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

```
>ID.6
CIEEVL
 >ID.7
KGDYRSK
 >ID.8
GPEELRLM
 >ID.9
IKVSQQEEK
 >ID.10
KWPCDQGCWS
 >ID.11
WNQPLIVNWVW
 >ID.12
YQYWYSHTCCEL
```

> Q.Are any of these sequences unique in nature- i.e. never found in nature. We can search “refseq-protein” and look for 100% ID and 100% coverage matches with BLASTp

All of these sequences have matches, and can be found in nature. Based on the searches of “refseq-protein”, the searches have 100% ID and 100% coverage mathes with BLASTp, indicating that they can be found in nature.