

BIMM 143 Class 06

Xaler Lu (A17388454)

Background

All functions in R have at least three things:

- A **name** that we use to call the function
 - One or more input **arguments**
 - The **body** the lines of R code that do the work
- ```
function(arg1,arg2) { paste(body1,body2) }
```

## The Add Function

Let's write a function called `add()` to sum up some numbers

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

Now we can use this function

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

```
[1] 101
```

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

```
[1] 101 2 101
```

```
add(x=10, y=10)
```

```
[1] 20
```

Q. What happens when we give a multiple element vector to both x and y?

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

```
[1] 200 2 200
```

Q. What if I give three inputs to the functions like z?

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

Q. What if I give one input to the function instead?

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

Let's make a new `add()` function that overrides y

```
addnew <- function(x, y=1) {
 x + y
}
```

```
addnew(c(100,1))
```

```
[1] 101 2
```

If we write our function with input arguments having no default value, then the user will be required to set them when they use the function. We can give our input argument default values by setting them equal to some sensible value - e.g. `y=1` in the `addnew()` function

## The Sequence Generating Function

The `sample()` will let us create a function to generate a nucleotide sequence

```
sample(1:10, size=4)
```

```
[1] 5 1 3 4
```

Q. Can I generate nine random numbers taken from the input vector `x=1:10`?

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

```
[1] 8 1 9 6 2 10 4 3 5
```

Q. Can I generate more than ten random numbers from `x=1:10`?

We can set `replace` to `TRUE`

```
sample(1:10, size=12, replace = TRUE)
```

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

Q. Write code for the `sample()` function that will generate nucleotide sequences of 6 bp.

```
sequence.practice <- sample(c("A","G","C","T"), size = 6, replace = TRUE)
paste(sequence.practice, sep="")
```

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

Q. Now we have our snippet, write a function `generate_dna()` that returns a user specified length DNA sequence:

```
generate_dna <- function(b=100) {
 a <- sample(c("A","G","C","T"), size = b, replace = TRUE)

 cat("Single-element vector output")
 paste(a, collapse = "")
}
generate_dna(100)
```

Single-element vector output

```
[1] "TCGACAGATCCCGTCATACGCGGCCAAATATGACCTTTTCGCTGGTAACCGTTTTCTGCTGTGTTGTGTTACCTTTACCGGACCGG"
```

We set `b=100` as the default if we set no values in `generate_dna()`

To remove the quotation marks in between each nucleotides, we would need to use `paste(x, collapse = "")` where `x` is the thing we want to paste.

Flow control means where the R brain goes in the code

```

good_mood <- F

if(good_mood) {
 cat("Great!")
} else {
 cat("Bummer!")
}

```

Bummer!

## Protein Generating Function

Q. Write a function that generates a user specified length protein sequences

```

generate_protein <- function(x) {
 a <- sample(c("A", "R", "N", "D", "C", "E", "Q", "G", "H", "I", "L", "K", "M", "F", "P",
 paste(a, collapse = "")
}

```

Q. Use that function to generate random protein sequences between length 6 and 12.

```

for(i in 6:12) {
 #FASTA ID Line
 cat(">", i, sep="", "\n")
 #Protein Sequence line
 cat(generate_protein(i), "\n")
}

```

```

>6
HWPSCQ
>7
SDQPIPM
>8
LPPNSIVP
>9
APIRFYDVD
>10
YFRLRECHFW
>11

```

NRRQRISPRAV  
>12  
HDTDFQSPHGEQ

Q. Are any of your sequences unique i.e. not found anywhere in nature?

Most of these sequences are found in nature according to BLAST except for sequences 10 and 12 with percent identity 88.89% and 81,82%, respectively.