# Class 6: R Functions

Grace Wang (PID: A16968688)

## Table of contents



### Function basics

First silly function - add some numbers:

Every R function has 3 components:

- name
- input argument(s), separated by commas
- body

```
add <- function(x, y = 10, z = 0){
  x + y + z
}
```

Once run, I can use this function like any other function.

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

```
[1] 101
```

```
add(1:4, 100)
```

```
[1] 101 102 103 104
```

```
add(1)
```

```
[1] 11
```

Functions can have "required" input arguments and "optional" input arguments. The optional arguments are defined with `= [default]` in the definition of the function.

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

```
[1] 111
```

### Generate a DNA sequence

> Write a function to return a DNA sequence of a user-specified length. Call it `generate_dna()`.

The `sample()` function can help.

```
bases <- c("A", "C", "G", "T")
sample(bases, size = 10, replace = T)
```

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

Now I have a working snippet. I can put this in the body of my function.

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

generate_dna()
```

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

```
generate_dna(100)
```

```
  [1] "C" "A" "A" "C" "C" "C" "C" "G" "A" "T" "T" "C" "A" "C" "G" "T" "C" "T"
 [19] "G" "C" "G" "G" "G" "T" "C" "A" "C" "A" "A" "T" "G" "T" "A" "T" "G" "C"
 [37] "A" "C" "G" "A" "A" "C" "G" "C" "T" "T" "A" "G" "G" "T" "T" "T" "A" "A"
 [55] "A" "C" "A" "G" "G" "T" "A" "T" "C" "C" "A" "A" "C" "G" "A" "G" "T" "T"
 [73] "T" "C" "T" "A" "T" "A" "C" "C" "A" "T" "C" "T" "G" "A" "A" "C" "T" "A"
 [91] "A" "G" "A" "G" "G" "C" "G" "A" "T" "G"
```

I want the ability to return a single sequence like "AGTACCTG" instead of many strings in a vector.

```
generate_dna <- function(length = 5, together = T){
  bases <- c("A", "C", "G", "T")
  sequence <- sample(bases, length, replace = T)
  if(together){
    sequence <- paste(sequence, collapse = "")
  }
  return(sequence)
}

generate_dna()
```

```
[1] "AGGAG"
```

```
generate_dna(, together = F)
```

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

### Generate a protein sequence

> Write a function to return protein sequences of different lengths and test whether these sequences are unique in nature.

We can get the set of 20 canonical amino acids from the **bio3d** package.

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

> Write a protein sequence generating function that will return sequences of a user-specified length.

```
generate_protein <- function(length = 10, string = T){
  aa <- bio3d::aa.table$aa1[1:20]
  aasequence <- sample(aa, size = length, replace = T)
  if(string == T){
    aasequence <- paste(aasequence, collapse = "")
  }
  return(aasequence)
}

generate_protein(20)
```

```
[1] "TFVKGIYFFLCTIYLYPIMI"
```

Generate random protein sequences of length 6-12 amino acids.

```
#generate_protein(6:12)
```

We can't use vectors as inputs. We can fix this by editing the function itself or by using the R **apply** family of functions.

```
sapply(6:12, generate_protein)
```

```
[1] "LKDCMV"       "WNVPTPR"      "DHDRDMDN"     "WAKIWKGVK"    "VDRSRHKFYC"
[6] "GCWMIRAWQAM"  "NAISTSDHCSMN"
```

It would be useful to put these in FASTA format output.

```
seqs <- sapply(6:12, generate_protein)
seqs
```

```
[1] "FCLEAI"       "VPIITDG"      "DDENCGFP"     "FEMFRWEFC"    "NFNLIQWKPK"
[6] "PHKQCQIKGVD"  "RYSKCPRHRSTM"
```

```
ids <- paste(">ID.", 6:12, sep = "")
fasta <- paste(ids, seqs, sep = "\n")
cat(fasta, sep = "\n")
```

```
>ID.6
FCLEAI
>ID.7
VPIITDG
>ID.8
DDENCGFP
>ID.9
FEMFRWEFC
>ID.10
NFNLIQWKPK
>ID.11
PHKQCQIKGVD
>ID.12
RYSKCPRHRSTM
```

> Determine if these sequences can be found in nature or are they unique? Why or why not?

This can be done using a blastp search. If both percent identity and query cover are 100%, the sequence is not unique and is found in nature. If the top hit has a value of <100% for either percent identity or query cover, then the sequence is not found in nature. For my sequences, lengths 6 and 7 were found in nature but 8-12 were unique.

Random sequences of length >=9 are unique and can't be found in databases. As such, protein sequences of length 9 or longer can generally be used to uniquely identify proteins.