Class 6: R functions

Peter Sax

Table of contents

Let's start writing our first silly function to add some numbers:

Every R function has 3 things:

- name (we get to pick this)
- input arguments (there are loads of these separated by a comma)
- the body (the R code that does the work)

1. Generate `add` function

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

I can just use this function

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

```
[1] 101
```

```
add(c(1,2,3,4), 100)
```

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

We can set a default for x or y so that we can run the function with only one argument

```
adddefault <- function(x, y=10){x + y}
adddefault(1)
```

```
[1] 11
```

```
adddefault(1, 100)
```

```
[1] 101
```

2. Generate DNA Sequence

> Q. Write a function to return a nucleotide sequence of a user specified length? Call it `generate_dna()`

```
generate_dna <- function(size){

  nucleotides <- c("A","T","C","G")
  sample(nucleotides, size, replace = TRUE)
}

generate_dna(5)
```

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

```
generate_dna(20)
```

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

I want the ability to return a sequence like "AGTACCTG" where the result is only one element.

```
generate_dna2 <- function(size, together = TRUE){
  nucleotides <- c("A","T","C","G")
  sequence <- sample(nucleotides, size=size, replace = TRUE)
  if(together) {sequence <- paste(sequence, collapse="")}
  return(sequence)
}
```

```
generate_dna2(10)
```

```
[1] "GAGTGTTACA"
```

```
generate_dna2(10, together=FALSE)
```

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

3. Generate protein sequence

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

```
#install.packages("bio3d")
```

> Q. Write a function, `generate_protein()` to return protein sequences of a user specified length

```
generate_protein1 <- function(size=5, together = TRUE){
  aminos <- c("A", "R", "N", "D", "C", "Q", "E", "G", "H", "I", "L", "K", "M", "F", "P", "S"
  protseq <- sample(aminos, size=size, replace=TRUE)
  if(together){protseq <- paste(protseq, collapse = "")}
  return(protseq)
}
```

```
generate_protein1(11)
```

```
[1] "ICTAVYWEPMK"
```

> Q. Write a function that generates sequences of length 6 to 12.

```
sapply(6:12, generate_protein1)
```

```
[1] "AASFCN"       "MEGDTNI"      "TWGMYLYK"     "YMVSDDVTV"    "GLTMWTSMHR"
[6] "LQTFYFIIYNF"  "QFEDCREWFAPH"
```

It would be cool and useful if I could get FASTA format output.

```r
ans <- sapply(6:12, generate_protein1)
ans
```

```
[1] "IKSEWV"       "KTIAPDA"      "DSFMASKG"     "NHIAYLGSD"    "ECLEQMLLMC"
[6] "PLCTTDIYGNH"  "WFLIAVFLFRQT"
```

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

```
>ID.6
IKSEWV
>ID.7
KTIAPDA
>ID.8
DSFMASKG
>ID.9
NHIAYLGSD
>ID.10
ECLEQMLLMC
>ID.11
PLCTTDIYGNH
>ID.12
WFLIAVFLFRQT
```

```r
id.line <- paste(">ID.", 6:12, sep = "")
seq.line <- paste(id.line, ans, sep = "\n")
cat(seq.line, sep = "\n")
```

```
>ID.6
IKSEWV
>ID.7
KTIAPDA
>ID.8
DSFMASKG
>ID.9
NHIAYLGSD
>ID.10
ECLEQMLLMC
>ID.11
PLCTTDIYGNH
>ID.12
WFLIAVFLFRQT
```

Q. Determine if these sequences can be found in nature or if they are unique.

After using a BLASTp search by inputting the FASTA code, I determined that length 9, 10, 11, 12 are unique, but 6, 7, 8 are not because there are sequences with 100% coverage and identity.