Class06: function

Kate Zhou (PID:A17373286)

Table of contents

1.Add function

New function - name (function name) - input arguments (there can be loads of these seperated by a comma) - the body (the R code that deos the work)

```r
add <- function(x, y=10, z = 100) {
  x+y+z
}
```

I can just use this funtion like any functionas long as R knows about it (i.e. run the code chunk)

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

```
[1] 201
```

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

```
[1] 201 202 203 204
```

```r
add(1)
```

```
[1] 111
```

Functions can have “required” input arguments and “optional” input arguments. The optional arguments are defined with an equals default value (`y=100`) in the function definition

2. Generate DNA Function

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

```r
students <- c("jeff", "jeremy", "peter")

sample(students, size=5, replace=TRUE)
```

```
[1] "jeff"   "peter"  "jeff"   "jeremy" "peter"
```

Now work with `bases` rather than `students`

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

sample(bases, size=10, replace=TRUE)
```

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

Now I have a working “snipt” of

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

```r
generate_dna()
```

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

I want the ability to return a sequence like “AGCACCTG” (i.e a one element vector where the bases are all together)

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

```r
generate_dna()
```

```
[1] "CAGTT"
```

3. Generate Protein Function

> Q. Write a protein sequence generating function that will return sequence of a user specified length?

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

```r
library("bio3d")
bio3d::aa.table$aa1[1:20]
```

```
 [1] "A" "R" "N" "D" "C" "Q" "E" "G" "H" "I" "L" "K" "M" "F" "P" "S" "T" "W" "Y"
[20] "V"
```

```r
generate_protein <-function(size=6, together=TRUE) {
  bases <- bio3d::aa.table$aa1[1:20]

  sequence <- sample(bases, size=size, replace=TRUE)
  if (together) {
    sequence <- paste(sequence, collapse = "")
  }
  return(sequence)
}
```

We can fix this inability to generate multiple sequences by either editing and adding to the functio body code (e.g. a for loop) or by using the R **apply** family of utility functions

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

```
[1] "EETHTE"       "ADKTFDS"      "GMRSLQWY"     "VSFHSEWRG"    "GHHPQFTEPM"
[6] "LIRLCDRDGYH"  "GKVHRQLAVRNI"
```

```r
generate_protein()
```

```
[1] "QRQSRR"
```

Q. Generate random protein sequences of length 6 to 12 amino acids.

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

cat(ans, sep="\n")
```

```
EVKAPT
MWRSIRT
NGMIHTCY
AFNMNYHTA
TTEEWEQGQP
WGKAPGHEICD
LAQYELKEIVTF
```

I want this to look like

```
>ID.6
EWIGVH
>ID.7
IHGPGHY
>ID.8
GQMKQNHS
>ID.9
WTVWYSTGI
>ID.10
TQSKKAGTRD
>ID.11
EVHTIMNENIG
>ID.12
FHYFDYYHIGAW
```

```r
id_head <- function(number) {
  paste(">ID.", as.character(number), sep="")
}
proteins <- sapply(6:12, generate_protein)
heads <- sapply(6:12, id_head)
cat(paste(heads, proteins, sep="\n"), sep = "\n")
```

```
>ID.6
VAIYCR
>ID.7
VHFRETI
>ID.8
CMKRAGWS
>ID.9
HGVEKDSPD
>ID.10
FRKSRIMRDI
>ID.11
SIYIVGRRHQT
>ID.12
VGRTVIDKLFYA
```

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

```
>ID.6
EVKAPT
>ID.7
MWRSIRT
>ID.8
NGMIHTCY
>ID.9
AFNMNYHTA
>ID.10
TTEEWEQGQP
>ID.11
WGKAPGHEICD
>ID.12
LAQYELKEIVTF
```

> Q. Determin if these sequences can be found in nature It would be cool and useful if I could get FASTA format output

I BLASTp searched my FASTA format sequences against NR and found that length 6, 7 are not unique and can be found in the databases with 100% coverage and 100% identity.

But for length 8, the coverage is 88% with 100% identity, or 100% coverage and 87.50% identity. For length 9, 100% coverage and 89% identity.

Random sequences of length 8 and above are unique and cannot be found in database.