

Actividad 2

Análisis de biología computacional BT1013.525

Bryan Manuel De la O Perea A01246337

Andrés Sarellano Acevedo A01245418

Maximiliano Villegas García A01635825

Víctor Manuel Puga Ruiz A01568636

Helpers

```
dna.bases <- c("A", "T", "C", "G")
rna.bases <- c("A", "U", "C", "G")

dna.clean <- function(dna) {
  vec <- gsub("[53' -]", "", dna) # remove 5 3 ' - and spaces
  dna.str(vec)
}

dna.validate <- function(dna) {
  str <- dna.clean(dna)
  test <- gsub("[ATCGN]", "", str)
  stopifnot(test == "")
}

rna.validate <- function(dna) {
  str <- dna.clean(dna)
  test <- gsub("[AUCGN]", "", str)
  stopifnot(test == "")
}

dna.vec <- function(dna.s) {
  clean <- dna.clean(dna.s)
  strsplit(clean, "")[[1]]
}

dna.str <- function(dna.v) {
  stopifnot(is.vector(dna.v))
  paste(dna.v, collapse="")
}

dna.forward <- function(dna) {
  v <- strsplit(dna, "")[[1]]
  v[1] != "3"
}

dna.reverse.labels <- function(dna) {
  is.forward <- dna.forward(dna)
```

```
beginning <- if (is.forward) "3'" else "5'"
ending <- if (is.forward) "-5'" else "-3'"

c(beginning, ending)
}
```

1

```
dna.rand <- function(n) {
  result <- sample(dna.bases, n, replace = TRUE)
  paste("5'", dna.str(result), "-3'", sep = "")
}

dna.seq <- dna.rand(12)

print(dna.seq)

## [1] "5'-ACAATGCGTCAA-3'"
```

2

```
dna.length <- function(dna) {
  vec <- dna.vec(dna)
  length(vec)
}

print(dna.length("ATGGCC"))

## [1] 6
print(dna.length("5'-ATGGCC-3'"))

## [1] 6
print(dna.length("NNGGCC"))

## [1] 6
print(dna.length("ATG GCC"))

## [1] 6
print(dna.length("ATG-GCC"))

## [1] 6
```

3

```
dna.percent <- function(dna) {
  l <- dna.length(dna)
  total <- data.frame(
    "A" = 0,
    "T" = 0,
    "C" = 0,
    "G" = 0,
    "N" = 0
```

```
)

for (base in dna.vec(dna)) {
  total[base] <- total[base] + 1
}
result <- apply(total, 1, function(x) { x/1 * 100 } )
result
}

print(dna.percent("ATGGCC"))
```

```
##      [,1]
## A 16.66667
## T 16.66667
## C 33.33333
## G 33.33333
## N  0.00000
```

```
print(dna.percent("5'-ATGGCC-3'"))
```

```
##      [,1]
## A 16.66667
## T 16.66667
## C 33.33333
## G 33.33333
## N  0.00000
```

```
print(dna.percent("NNGGCC"))
```

```
##      [,1]
## A  0.00000
## T  0.00000
## C 33.33333
## G 33.33333
## N 33.33333
```

```
print(dna.percent("ATG GCC"))
```

```
##      [,1]
## A 16.66667
## T 16.66667
## C 33.33333
## G 33.33333
## N  0.00000
```

```
print(dna.percent("ATG-GCC"))
```

```
##      [,1]
## A 16.66667
## T 16.66667
## C 33.33333
## G 33.33333
## N  0.00000
```

4

```
dna.to.rna <- function(dna) {
  stopifnot(dna.forward(dna))
  dna.validate(dna)

  matches <- list(
    "A"="U",
    "T"="A",
    "C"="G",
    "G"="C",
    "N"="N"
  )
  result <- sapply(dna.vec(dna), function(b) { matches[[b]] })
  paste("3'", dna.str(result), "-5'", sep = "")
}
```

```
cat("DNA      ", "ATGGCC", "\n",
    "RNA      ", dna.to.rna("ATGGCC"), "\n\n", sep="")
```

```
## DNA      ATGGCC
## RNA      3'-UACCGG-5'
```

```
cat("DNA      ", "5'-ATGGCC-3", "\n",
    "RNA      ", dna.to.rna("5'-ATGGCC-3"), "\n", sep="")
```

```
## DNA      5'-ATGGCC-3
## RNA      3'-UACCGG-5'
```

5

```
rna.to.protein <- function(rna) {
  is.divisible <- (dna.length(rna) %% 3) == 0
  stopifnot(is.divisible)
  rna.validate(rna)

  matches <- list(
    "GCA"="Ala", "GCC"="Ala", "GCG"="Ala", "GCU"="Ala",
    "UGC"="Cys", "UGU"="Cys",
    "AAC"="Asp", "AAU"="Asp",
    "GAA"="Glu", "GAG"="Glu",
    "UUC"="Phe", "UUU"="Phe",
    "GGA"="Gly", "GGC"="Gly", "GGG"="Gly", "GGU"="Gly",
    "CAC"="His", "CAU"="His",
    "AUA"="Ile", "AUC"="Ile", "AUU"="Ile",
    "AAA"="Lys", "AAG"="Lys",
    "CUA"="Leu", "CUC"="Leu", "CUG"="Leu", "CUU"="Leu",
    "UUA"="Leu", "UUG"="Leu",
    "AUG"="Met",
    "CCA"="Pro", "CCC"="Pro", "CCG"="Pro", "CCU"="Pro",
    "CAA"="Gln", "CAG"="Gln",
    "AGA"="Arg", "AGG"="Arg", "CGA"="Arg", "CGC"="Arg", "CGG"="Arg", "CGU"="Arg",
    "AGC"="Ser", "AGU"="Ser", "UCA"="Ser", "UCC"="Ser", "UCG"="Ser", "UCU"="Ser",
    "ACA"="Thy", "ACC"="Thy", "ACG"="Thy", "ACU"="Thy",
```

```

    "GUA"="Val", "GUC"="Val", "GUG"="Val", "GUU"="Val",
    "UGG"="Try", "UAC"="Tyr", "UAU"="Tyr",
    "UAA"="STOP", "UAG"="STOP", "UGA"="STOP"
  )
  plain <- dna.clean(rna)
  codons <- strsplit(plain, "(?<=.{3})", perl = TRUE)[[1]]

  prot <- ""
  for (codon in codons) {
    amino <- matches[[codon]]
    if (amino == "STOP") {
      break
    } else {
      prot <- paste(prot, amino, " ", sep="")
    }
  }
  prot
}

cat("RNA      ", "3'-AUG CUU ACU GGC-5'", "\n",
    "Protein   ", rna.to.protein("3'AUG CUU ACU GGC-5'"), "\n\n", sep="")

## RNA      3'-AUG CUU ACU GGC-5'
## Protein   Met Leu Thy Gly

cat("RNA      ", "AUG CUU ACU GGC", " (3'-5')\n",
    "Protein   ", rna.to.protein("AUG CUU ACU GGC"), "\n\n", sep="")

## RNA      AUG CUU ACU GGC (3'-5')
## Protein   Met Leu Thy Gly

cat("RNA      ", "3'-AUG CUU ACU GGC UUG AAA UAA AAA AAA AAA-5'", "\n",
    "Protein   ", rna.to.protein("3'AUG CUU ACU GGC UUG AAA UAA AAA AAA AAA-5'"),
    "(UAA is a stop codon)", "\n\n", sep="")

## RNA      3'-AUG CUU ACU GGC UUG AAA UAA AAA AAA AAA-5'
## Protein   Met Leu Thy Gly Leu Lys (UAA is a stop codon)

```

6

```

dna.reverse <- function(dna) {
  labels <- dna.reverse.labels(dna)

  vec <- dna.vec(dna)
  vec <- rev(vec)

  paste(labels[1], dna.str(vec), labels[2], sep = "")
}

cat("Direct   ", "5'-ATGGCC-3'", "\n",
    "Reverse   ", dna.reverse("ATGGCC"), "\n\n", sep="")

## Direct   5'-ATGGCC-3'

```

```
## Reverse 3'-CCGGTA-5'
cat("Direct   ", "3'-ATGGCC-5'", "\n",
    "Reverse   ", dna.reverse("3'-ATGGCC-5'"), "\n\n", sep="")

## Direct   3'-ATGGCC-5'
## Reverse  5'-CCGGTA-3'
cat("Direct      ", "ATGGCC", "   (5'-3')", "\n",
    "Reverse     ", dna.reverse("ATGGCC"), "\n", sep="")

## Direct      ATGGCC   (5'-3')
## Reverse  3'-CCGGTA-5'
```

7

```
dna.complement <- function(dna) {
  stopifnot(dna.forward(dna))
  labels <- dna.reverse.labels(dna)
  matches <- list(
    "A"="T",
    "T"="A",
    "C"="G",
    "G"="C"
  )
  result <- sapply(dna.vec(dna), function(b) { matches[[b]] })
  paste(labels[1], dna.str(result), labels[2], sep = "")
}

cat("Direct      ", "5'-ATGGCC-3'", "\n",
    "Complement   ", dna.complement("ATGGCC"), "\n\n", sep="")

## Direct      5'-ATGGCC-3'
## Complement  3'-TACCGG-5'
cat("Direct      ", "ATGGCC", "   (5'-3')", "\n",
    "Complement   ", dna.complement("ATGGCC"), "\n", sep="")

## Direct      ATGGCC   (5'-3')
## Complement  3'-TACCGG-5'
```

8

```
dna.reverse.complement <- function(complement) {
  dna.reverse(complement)
}

comp <- dna.complement("ATGGCC")
cat("Complement    ", comp, "\n",
    "Inverse Comp.  ", dna.reverse.complement(comp), "\n", sep="")

## Complement    3'-TACCGG-5'
## Inverse Comp. 5'-GGCCAT-3'
```