Exercise-1

Ramesh Chellaton

2023-01-24

R Markdown

Load libraries here

```
eval = TRUE
message = FALSE
warning = FALSE
echo = FALSE
invisible(library(seqinr))
invisible(library(dplyr))
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:seqinr':
##
##     count

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
invisible(library(Biostrings))
```

```
## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:seqinr':
##
##     translate

## The following object is masked from 'package:base':
##
##     strsplit
```

Data load: Read 2 files - dengue.fasta and leprae.fasta (Mycobacterium leprae)

```
leprae <- read.fasta('leprae.fasta')
dengue <- read.fasta('dengue.fasta')
```

```
leprae_seq <- leprae[[1]]
dengue_seq <- dengue[[1]]
```

Setup: convert fasta list files into sequence files that Biostrings and seqinr can use

```
start <- length(dengue_seq)-20
end <- length(dengue_seq)
print(dengue_seq[start:end])
```

Q1. Last 20 nucleotdies of dengue genome

```
##  [1] "g" "c" "t" "g" "t" "t" "g" "a" "a" "t" "c" "a" "a" "c" "a" "g" "g" "t" "t"
## [20] "c" "t"
```

```
print(paste('Length of Mycobacterium leprae in nucleotides:',
            length(leprae_seq)))
```

Q2. Length in nucleotides of Mycobacterium leprae

```
## [1] "Length of Mycobacterium leprae in nucleotides: 348450"
```

```
print(seqinr::count(leprae_seq,1))
```

Q3. Number of 'A','C','G','T' bases in Mycobacterium leprae:

```
##
##      a      c      g      t
##  67024  95349 108367  77710
```

```
GC(leprae_seq)
```

Q4. GC Content of Mycobacterium leprae:

```
## [1] 0.5846348
```

```
seqinr::count(comp(leprae_seq),1)
```

Q5. Number of 'a','c','g','t' bases in Mycobacterium leprae complement:

```
##
##      a      c      g      t
##  77710 108367  95349  67024
```

```
dimer_tbl <- seqinr::count(leprae_seq, 2)
dimer_tbl[c('cc','gc','cg')]
```

Q6. Number of 'CC','CG','GC' DNA words in Mycobaterium leprae:

```
##
##    cc    gc    cg
## 23052 31920 33508
```

Q7. Number of 'cc','cg','gc' in the first 1000 and last 1000 of leprae seq: Q7.1: from 1:1000 positions:

```
start <- 1
end <- 1000
dimer_tbl <- seqinr::count(leprae_seq[start:end], 2)
dimer_tbl[c('cc','cg','gc')]
```

```
##
##  cc  cg  gc
##  92 105 100
```

Q7.2 from 348450-1000: 348450

```
end <- length(leprae_seq)
start <- end - 1000
dimer_tbl <- seqinr::count(leprae_seq[start:end], 2)
dimer_tbl[c('cc','cg','gc')]
```

```
##
##  cc  cg  gc
##  53 109  79
```