Analysis

Corbin Montminy

2023-02-13

[My Github] (https://github.com/CBMontminy (https://github.com/CBMontminy))

Importing Data

```
Sequences=read.csv("Sequences.csv")
```

```
Sequences$Sequence
```

```
## [1] "AGCATGCAAGTCAAACGAGATGTAGCAATACATCTAGTGGCGAACGGGTGAGTAACGCGTGGATGATCTACCTATGAGATGGGGATAACTATTAGAAATAGTAGCTAATACCGAAT
AAGGTCAATTAATTTGTTAATTGATGAAAGGAAGCCTTTAAAGCTTCGCTTGTAGATGAGTCTGCGTCTTATTAGTTAGTTGGTAGGGTAAATGCCTACCAAGGCGATGATAAGTAACCGGCCT
GAGAGGGTGAACGGTCACACTGGAACTGAGACACGGTCCAGACTCCTACGGGAGGCAGCAGCTAAGAATCTTCCGCAATGGGCGAAAGCCTGACGGAGCGACACTGCGTGAATGAAGAAGGTCG
AAAGATTGTAAAATTCTTTTATAAATGAGGAATAAGCTTTGTAGGAAATGACGAAGTGATGACGTTAATTTATGAATAAGCCCCGGCTAATTACGTGCCAGCAGCCGCGGTAATACG"
## [2] "AGCATGCAAGTCAAACGGGATGTAGCAATACATTCAGTGGCGAACGGGTGAGTAACGCGTGGATGATCTACCTATGAGATGGGGATAACTATTAGAAATAGTAGCTAATACCGAAT
AAGGTCAGTTAATTTGTTAATTGATGAAAGGAAGCCTTTAAAGCTTCGCTTGTAGATGAGTCTGCGTCTTATTAGCTAGTTGGTAGGGTAAATGCCTACCAAGGCAATGATAAGTAACCGGCCT
GAGAGGGTGAACGGTCACACTGGAACTGAGATACGGTCCAGACTCCTACGGGAGGCAGCAGCTAAGAATCTTCCGCAATGGGCGAAAGCCTGACGGAGCGACACTGCGTGAATGAAGAAGGTCG
AAAGATTGTAAAATTCTTTTATAAATGAGGAATAAGCTTTGTAGGAAATGACAAAGTGATGACGTTAATTTATGAATAAGCCCCGGCTAATTACGTGCCAGCAGCAGCGGTAATACG"
## [3] "AGCATGCAAGTCAAACGAGATGTAGTAATACATCTAGTGGCGAACGGGTGAGTAACGCGTGGATGATCTACCTATGAGATGGGGATAACTATTAGAAATAGTAGCTAATACCGAAT
AAGGTCAATTAATTTGTTAATTGATGAAAGGAAGCCTTTAAAGCTTCGCTTGTAGATGAGTCTGCGTCTTATTAGTTAGTTGGTAGGGTAAATGCCTACCAAGGCGATGATAAGTAACCGGCCT
GAGAGGGTGAACGGTCACACTGGAACTGAGACACGGTCCAGACTCCTACGGGAGGCAGCAGCTAAGAATCTTCCGCAATGGGCGAAAGCCTGACGGAGCGACACTGCGTGAATGAAGAAGGTCG
AAAGATTGTAAAATTCTTTTATAAATGAGGAATAAGCTTTGTAGGAAATGACGAAGTGATGACGTTAATTTATGAATAAGCCCCGGCTAATTACGTGCCAGCAGCCGCGGTAATACG"
```

Creating a for loop to count the number of nucleotides

```
for(i in 1:3){
  print(paste("Number of A's in sequence", i, "-", nchar(gsub("A", "", Sequences$Sequence[i]))))
  print(paste("Number of T's in sequence", i, "-", nchar(gsub("T", "", Sequences$Sequence[i]))))
  print(paste("Number of G's in sequence", i, "-", nchar(gsub("G", "", Sequences$Sequence[i]))))
  print(paste("Number of C's in sequence", i, "-", nchar(gsub("C", "", Sequences$Sequence[i]))))
}
```

```
## [1] "Number of A's in sequence 1 - 327"
## [1] "Number of T's in sequence 1 - 367"
## [1] "Number of G's in sequence 1 - 350"
## [1] "Number of C's in sequence 1 - 399"
## [1] "Number of A's in sequence 2 - 326"
## [1] "Number of T's in sequence 2 - 367"
## [1] "Number of G's in sequence 2 - 350"
## [1] "Number of C's in sequence 2 - 400"
## [1] "Number of A's in sequence 3 - 327"
## [1] "Number of T's in sequence 3 - 366"
## [1] "Number of G's in sequence 3 - 350"
## [1] "Number of C's in sequence 3 - 400"
```

```
Seq1=c(327, 367, 350, 399)
Seq2=c(326, 367, 350, 400)
Seq3=c(327, 366, 350, 400)
SeqTab=rbind(Seq1, Seq2, Seq3)
colnames(SeqTab)=c("A's", "T's", "G's", "C's")
rownames(SeqTab)=c("Sequence 1", "Sequence 2", "Sequence 3")
SeqTab
```

```
##            A's T's G's C's
## Sequence 1 327 367 350 399
## Sequence 2 326 367 350 400
## Sequence 3 327 366 350 400
```

Image Link (https://www.bayarealyme.org/about-lyme/what-causes-lyme-disease/borrelia-burgdorferi/)

Wikipedia Link to Borrelia burgdorferi (https://en.wikipedia.org/wiki/Borrelia_burgdorferi)

GC Content

Calculate GC content

```
GC=paste((SeqTab[,4]+SeqTab[,3])/(SeqTab[,1]+SeqTab[,2]+SeqTab[,3]+SeqTab[,4])*100)
```

Creating a dataframe to output a table

```
ID=gsub(">(HQ.*\\.1).*", "\\1", Sequences$Name)
GC=c((350+399)/(327+367+350+399)*100, (350+400)/(326+367+350+400)*100, (350+400)/(327+366+350+400)*100)
GC=round(GC, digits=2)
GCCont=data.frame(ID, GC)
colnames(GCCont)=c("ID", "GC Content (%)")
print(GCCont)
```

```
##           ID GC Content (%)
## 1 HQ433692.1          51.91
## 2 HQ433694.1          51.98
## 3 HQ433691.1          51.98
```

Part II

Human isolate, unknown sequence
GCCTGATGGAGGGGGATAACTACTGGAAACGGTAGCTAATACCGCATGACCTCGCAAGAGCAAAGTGGGGGACCT
TAGGGCCTCACGCCATCGGATGAACCCAGATGGGATTAGCTAGTAGGTGGGGTAATGGCTCACCTAGGCGACGAT
CCCTAGCTGGTCTGAGAGGATGACCAGCCACACTGGAACTGAGACACGGTCCAGACTCCTACGGGAGGCAGCAGT
GGGGAATATTGCACAATGGGCGCAA

Creating sequence object

```
UnknSeq="GCCTGATGGAGGGGGATAACTACTGGAAACGGTAGCTAATACCGCATGACCTCGCAAGAGCAAAGTGGGGGACCTTAGGGCCTCACGCCATCGGATGAACCCAGATGGGATTAGC
TAGTAGGTGGGGTAATGGCTCACCTAGGCGACGATCCCTAGCTGGTCTGAGAGGATGACCAGCCACACTGGAACTGAGACACGGTCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCAC
AATGGGCGCAA"
```

Loading packages

```
library(annotate)
```

```
## Loading required package: AnnotationDbi
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## 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: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Loading required package: IRanges
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'
```

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

```
##
## Attaching package: 'IRanges'
```

```
## The following object is masked from 'package:grDevices':
##
##     windows
```

```
## Loading required package: XML
```

Conducting BLAST search for the unknown sequence

```
Blast=blastSequences(UnknSeq, timeout=240, hitListSize=10, as='data.frame')
```

Loading BLAST object from Blaste.R script

```
Blast=read.csv("Blast.csv")
```

Results

Removing repetitve hits

```
Data=Blast[!duplicated(Blast$Hit_id),]
```

Producing tables

```
ID=Data$Hit_id
Score=Data$Hsp_score
Gap=Data$Hsp_gaps
Def=Data$Hit_def
Tab1=data.frame(ID, Score, Gap)
colnames(Tab1)=c("Hit ID", "Hit Score", "Gaps in Sequence")
```

```
Tab2=data.frame(ID,Def)
colnames(Tab2)=c("Hit ID", "Host")
print(Tab1)
```

```
##                          Hit ID Hit Score Gaps in Sequence
## 1  gi|2290367315|gb|CP096666.1|       500                0
## 2  gi|2279781810|gb|CP084343.1|       500                0
## 3  gi|2279769524|gb|CP084339.1|       500                0
## 4  gi|2279758382|gb|CP084336.1|       500                0
## 5  gi|2152215862|gb|CP071944.1|       500                0
## 6  gi|2029911040|gb|CP063303.2|       500                0
## 7  gi|2029910980|gb|CP064125.2|       500                0
## 8  gi|2029910731|gb|CP064119.2|       500                0
## 9  gi|2029910499|gb|CP064122.2|       500                0
## 10 gi|1948828809|gb|CP065918.1|       500                0
```

```
print(Tab2)
```

```
##                          Hit ID
## 1  gi|2290367315|gb|CP096666.1|
## 2  gi|2279781810|gb|CP084343.1|
## 3  gi|2279769524|gb|CP084339.1|
## 4  gi|2279758382|gb|CP084336.1|
## 5  gi|2152215862|gb|CP071944.1|
## 6  gi|2029911040|gb|CP063303.2|
## 7  gi|2029910980|gb|CP064125.2|
## 8  gi|2029910731|gb|CP064119.2|
## 9  gi|2029910499|gb|CP064122.2|
## 10 gi|1948828809|gb|CP065918.1|
##                                                      Host
## 1        Yersinia pestis EV76-CN chromosome, complete genome
## 2      Yersinia pestis strain 20 chromosome, complete genome
## 3      Yersinia pestis strain 94 chromosome, complete genome
## 4       Yersinia pestis strain R chromosome, complete genome
## 5          Yersinia pseudotuberculosis strain 598 chromosome
## 6     Yersinia pestis strain 14D chromosome, complete genome
## 7   Yersinia pestis strain M2085 chromosome, complete genome
## 8   Yersinia pestis strain C-792 chromosome, complete genome
## 9  Yersinia pestis strain M-1770 chromosome, complete genome
## 10      Yersinia pestis EV NIIEG chromosome, complete genome
```

As you can see, all of our 10 matches have a score of 500 with no gaps in the sequence, indicating 100% query coverage and that they are the exact matches.

From the host of the sequence matches, we get 9/10 for Yersinia pestis. The other 1/10 is for Yersinia pseudotuberculosis, a close relative of Y, pestis, indicating that this sequence may be conserved across species. Given the results, however, it is more likely that the unknown strand in the patient belongs to Yersinia pestis.

This indicates that the patient is likely suffering from one of the three forms of the plague due to an infection with Yersinia pestis, which is highly concerning, and should be dealt with immediately.

Yersinia pestis is a gram negative bacterium that can infect humans.Yarsinia pestis is the bacteria responsible for the Black Death and causes the plague. The forms of the plague can vary depending on location of infection. Pneumonic plague infects the lungs, the bubonic plague affects the lymph nodes, and the septicemic plague infects the blood.

The bubonic and septicemic forms of the plague can be transmitted to humans via the Oriental rat flea, while the pneumonic is typically spread between people via infectious droplets in the air.