Lab Assignment 2

STA 100 | A. Farris | Spring 2020

A pdf copy of your assignment is due at the end of the day (11:59pm PT) Monday, April 27. Submission of the pdf will be through Canvas. Please put in the effort to make it look reasonably professional – you're encouraged to use R Markdown. Note that specific tasks for you are highlighted. You are free to work in groups, but you must submit your own writeup, and run your own code.

Introduction to random sampling with sequencing

In this lab, we will briefly explore random sampling in the genomic context. In doing so we will simulate a hypothetical genetic test, and estimate its (laboratory) specificity.

A simple test

As an example of what we'll do: suppose that we want to identify whether a virus with the genome 'AGCGATTGAGTATGGATTTCAA' is present from a sample that may consist either of this sequence or of a different one.

We have the ability to read sequences of nucleotides, but we can only read some limited number (say eight) of consecutive nucleotides at a time. To check for the specific sequence above, then, what we could do is first simplify the problem by selecting a *reference* subsequence, say 'GATTGA', to check for. The presence of this sequence may not necessarily confirm the presence of the exact genetic sequence that we started with, but it does at least alert us to the possibility.

So, we would select a random subsequence of length 8 from the sample, and check it for the presence of our reference subsequence. If we select 'TATGGATT' as our test sequence, for example, we can check it and see that 'GATTGA' does not occur in it (in its entirety). Of course only checking once does not do a very good job of checking for the presence of our reference subsequence, so we could then repeat this many times, independenly selecting test subsequences to check each time for the presence of our reference subsequence. If we do this enough, the idea goes, then we can be confident of seeing the reference subsequence at least once, if it is present. But, how many times should we check these random test subsequences, in order to have a high probability of detecting the presence of the test subsequence? For example, does 10 times give us a high probability?

We can carry out a simulation to estimate this. Here is some R code which defines a function that will carry out the test for us:

```r
CheckOnce <- function(FullSequence,TestRef,readLength){
        # Randomly sample a location on the genome to read
        RandomIndex <- sample(1:(nchar(FullSequence) - readLength + 1),1)
        RandomSnippet <- substr(FullSequence, RandomIndex, RandomIndex + readLength - 1)
        # Check to see if it contains the ref. string
        grepl(TestRef,RandomSnippet)
}
Test <- function(FullSequence,TestRef,readLength,numReps){
        # Repeat the test numReps times;
        # if any find the reference, return +
        Reps <- replicate(numReps,CheckOnce(FullSequence,TestRef,readLength))
        ifelse(any(Reps),"+","-")
}
```

We can try out the test once:

```
Test("AGCGATTGAGTATGGATTTCAA","GATTGA",8,10) # this is how we use the function
```

```
[1] "-"
```

But is this result typical? Let's repeat this test a thousand times independently, and see in what proportion the result is positive. This will be our estimate for the sensitivity of the test.

```
repl1000 <- replicate(1000,Test("AGCGATTGAGTATGGATTTCAA","GATTGA",8,10))
results <- table(repl1000)
results
```

```
repl1000
  -   +
110 890
```

So we would estimate that the test has a specificity of about $\frac{890}{110+890} \approx 0.89$. So, if the sample actually consists of the sequence that we started with, we would estimate this to be the probability with which we get a positive test result.

In order to be clear, though, we should really call this the *laboratory* specificity, because it takes into account only uncertainty associated with the way that we are sequencing: we are not taking into account additional uncertainties that might arise in the clinical setting.

Finally, we could increase the accuracy of this estimate by using more than 1000 replications, at the cost of more computing time.

A test for COVID-19

Let's read into memory the reference genome[1] for the virus causing COVID-19:

```
SeqLines <- readLines("http://www.stat.ucdavis.edu/~affarris/CovidRef.txt")
CovSequence <- paste(SeqLines[-1], collapse="")
```

This leaves us with the object `CovSequence` in R, which is a character string identifying the nucleotides in the virus' genome. We can find out how many nucleotides are recorded using `nchar`:

```
nchar(CovSequence)
```

```
[1] 29903
```

So in this case the reference genome has 29903 nucleotides. We can look at small snippets of the sequence by using `substr`, for example to look at the first through the fiftieth nucleotides:

```
substr(CovSequence,1,50)
```

```
[1] "ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATCTC"
```

or we could look at the 20,000th through the 20,030th:

```
substr(CovSequence,20000,20030)
```

```
[1] "TTGATGGTCAAGTAGACTTATTTAGAAATGC"
```

Now let's create a hypothetical test. For our reference subsequence, let's use the subsequence from nucleotide 27202 to 27387 of the genome[2]:

```
RefSubseq <- substr(CovSequence,27202,27387)
```

For our test subsequences, let's use test sequences (reads) of length 3000. Finally, for each test, let's use 12 random reads.

[1] https://www.ncbi.nlm.nih.gov/nuccore/MN908947.3

[2] this subsequence encodes an ORF6 protein for the virus

```
repl1000 <- replicate(1000,Test(CovSequence,RefSubseq,3000,12))
results <- table(repl1000)
results
```

```
repl1000
  -   +
302 698
```

We would estimate that this test has a specificity of about $\frac{698}{302+698} \approx 0.7$.

If we were to increase the number of reads in the test from twelve, we could obtain a test with higher specificity. What would happen if we used a different reference subsequence?

Estimate the specificity of a test for COVID-19 that uses as its reference subsequence the subsequence from nucleotide 27894 to 28259 of the genome.[3] Use test sequences (reads) of length 3000, and for each test, use 12 random reads. Also use 1000 replications of the test, as before.

[3] this subsequence encodes an ORF8 protein for the virus

Appendix: R Script

```r
CheckOnce <- function(FullSequence,TestRef,readLength){
        # Randomly sample a location on the genome to read
        RandomIndex <- sample(1:(nchar(FullSequence) - readLength + 1),1)
        RandomSnippet <- substr(FullSequence, RandomIndex, RandomIndex + readLength - 1)
        # Check to see if it contains the ref. string
        grepl(TestRef,RandomSnippet)
}
Test <- function(FullSequence,TestRef,readLength,numReps){
        # Repeat the test numReps times;
        # if any find the reference, return +
        Reps <- replicate(numReps,CheckOnce(FullSequence,TestRef,readLength))
        ifelse(any(Reps),"+","-")
}
Test("AGCGATTGAGTATGGATTTCAA","GATTGA",8,10) # this is how we use the function
repl1000 <- replicate(1000,Test("AGCGATTGAGTATGGATTTCAA","GATTGA",8,10))
results <- table(repl1000)
results
SeqLines <- readLines("http://www.stat.ucdavis.edu/~affarris/CovidRef.txt")
CovSequence <- paste(SeqLines[-1], collapse="")
nchar(CovSequence)
substr(CovSequence,1,50)
substr(CovSequence,20000,20030)
RefSubseq <- substr(CovSequence,27202,27387)
repl1000 <- replicate(1000,Test(CovSequence,RefSubseq,3000,12))
results <- table(repl1000)
results
```