STA 6366, Homework.2

Carson Slater *Baylor University*

1. **Find a general formula for the expected value and variance of the number of times a word of length k shows up in a random string of letters of size N with an alphabet of size 4 without counting overlaps. You can assume that each nucleotide A,C,G,T is equally likely to show up at any point in the string N and that the nucleotide appearing at any position i is independent of that from any other position j. (Note, that not counting overlaps implies that if $w = ACA$, the string "ACACA" would count as only 1, but the string "ACACACA" would count as 2, since you can pull out two distinct copies of the pattern)**

Along a string size N, there are $N - k + 1$ windows where a word w of length k can show up. Let I_i, $i = 1, 2, \ldots, N - k + 1$ be the indicator function that w starts at position i. Let $Y_1(N) = \sum_{i=1}^{N-k+1} I_i$ be the total number of times that some word w of length k appears in a string of size N. We first want to find $\mathbb{E}[Y_1(N)]$. We assume an alphabet of size 4 with equally likely letters and do not count overlaps. Then, we have it that the probability of any k-letter word w is $\pi(w) = 4^{-k}$, and the number of available windows when not counting overlaps is $N - k + 1$. Essentially, each $I_i \overset{iid}{\sim} \text{ber}(4^{-k})$. So therefore

$$\begin{aligned}
\mathbb{E}[Y_1(N)] &= \mathbb{E}\left[\sum_{i=1}^{N-k+1} I_i\right] \\
&= \sum_{i=1}^{N-k+1} \mathbb{E}\left[I_i\right] \\
&= \sum_{i=1}^{N-k+1} \pi(w) \\
&= \frac{N-k+1}{4^k}.
\end{aligned}$$

Solving for $\text{Var}(Y_1(N))$, we have that the general formula for variance is,

$$\begin{aligned}
\text{Var}(Y_1(N)) &= \text{Var}\left(\sum_{i=1}^{N-k+1} I_i\right) \\
&= \sum_{i=1}^{N-k+1} \text{Var}(I_i) + 2\sum_{i=1}^{N-k} \sum_{j=i+1}^{N-k+1} \text{Cov}(I_i, I_j).
\end{aligned}$$

We know that for $j - i \geq k$, $\text{Cov}(I_i, I_j) = 0$. We also have that for $0 < j - i < k$, $\text{Cov}(I_i, I_j) = \mathbb{E}[I_i I_j] - \mathbb{E}[I_i]\mathbb{E}[I_j] = 0 - \pi(w)\pi(w) = -\pi(w)^2$. So then continuing on, we have that

$$\begin{aligned}
\mathrm{Var}(Y_1(N)) &= \sum_{i=1}^{N-k+1} \mathrm{Var}(I_i) + 2\sum_{i=1}^{N-k}\sum_{j=i+1}^{N-k+1} \mathrm{Cov}(I_i, I_j) \\
&= (N-k+1)(\pi(w))(1-\pi(w)) - 2\left((N-k+1)(k-1) - \underbrace{\frac{k(k-1)}{2}}_{\text{correcting for edges}}\right)\pi(w)^2 \\
&= (N-k+1)(\pi(w))(1-\pi(w)) - 2\left(N - \frac{3}{2}k + 1\right)(k-1)\pi(w)^2.
\end{aligned}$$

2. **Let $w = ACTAC$. For $N = 20, 100$ and $1,000$, calculate the mean and variance of the counts with and without overlaps. Comment on the differences between the variance for each level of N.**

N	Mean	Var (No Overlap)	Var (With Overlap)
20	0.015625	0.015507	0.015903
100	0.093750	0.092945	0.095783
1000	0.972656	0.964127	0.994431

3. **Using R (or python or something similar), write a version of the "FrequentWords" or "BetterFrequentWords" function that can group a k-mer with its reverse complement (e.g., if "ACCT" occurs 3 times and "AGGT" occurs 4 times, then the output for ACCT should be 7). Apply this function to the slightly edited E-coli ori (on canvas) and identify candidate(s) for the DnaA box using $k = 9$.**

```
# PROBLEM 3 ----------------------------------------------------------
reverse_complement <- function(pattern) {
  comp <- chartr("ACGT", "TGCA", pattern)
  paste(rev(strsplit(comp, "")[[1]]), collapse = "")
}

FrequentWordsWithRC <- function(text, k) {
  n <- nchar(text)
  freqMap <- list()

  # Build frequency map
  for (i in 1:(n - k + 1)) {
    pattern <- substr(text, i, i + k - 1)
    rc <- reverse_complement(pattern)
    key <- min(pattern, rc)
    freqMap[[key]] <- (freqMap[[key]] %||% 0) + 1
  }
```

```
  counts <- unlist(freqMap)
  maxCount <- max(counts)

  names(counts[counts == maxCount])
}

`%||%` <- function(a, b) if (!is.null(a)) a else b

path <- "/Users/carson/Documents/Baylor_Work/2026_Spring/Statistical_Bioinfor

dna_long <- read_lines(path) |>
  str_flatten()

result <- FrequentWordsWithRC(dna_long, 9)
cat("Most frequent k-mers (with RC grouping):", result[1:2], "\n", result[3:5]
```

Most frequent k-mers (with RC grouping): aaggatccg aggatccgg ataggtgtc aataggtgt caataggtg

4. **Let $X \sim \text{bin}(n, p)$, where $\mu = np$. Suppose $n \to \infty$ and $p \to 0$ such that np remains fixed at μ. Show that $X \xrightarrow{d} \text{Poi}(\mu)$. (Hint: use MGFs)**

Let $X \sim \text{bin}(n, p)$, where $\mu = np$. Suppose $n \to \infty$ and $p \to 0$ such that np remains fixed at μ. We show that $X \xrightarrow{d} \text{Poi}(\mu)$.

The MGF of $X \sim \text{bin}(n, p)$ is $M_X(t) = (1 - p + pe^t)^n$. Substituting $p = \mu/n$:

$$M_X(t) = \left(1 + \frac{\mu(e^t - 1)}{n}\right)^n.$$

As $n \to \infty$, we apply the standard limit $\lim_{n\to\infty}(1 + a/n)^n = e^a$ with $a = \mu(e^t - 1)$:

$$\begin{aligned}\lim_{n\to\infty} M_X(t) &= \lim_{n\to\infty}\left(1 + \frac{\mu(e^t - 1)}{n}\right)^n \\ &= e^{\mu(e^t-1)}.\end{aligned}$$

For $Y \sim \text{Poi}(\mu)$, the MGF is

$$M_Y(t) = e^{\mu(e^t-1)}.$$

Since $M_X(t) \to e^{\mu(e^t-1)}$ pointwise, and this limit is finite in an open neighborhood of $t = 0$, the continuity theorem for MGFs guarantees that $X \xrightarrow{d} \text{Poi}(\mu)$.

5. **Recall that we found the average number of contigs formed via shotgun sequencing has the following approximation:**

$$Ne^{-a}, \quad a = \frac{NL}{G}$$

If L and G are fixed, show that this is maximized when $N = G/L$.

We optimize by using calculus, differentiating Ne^{-a} with respect to N.

$$\begin{aligned}\frac{d}{dN}\left(Ne^{-a}\right) &= \frac{d}{dN}\left(Ne^{-NL/G}\right) \\ &= e^{-NL/G} + N \cdot \left(-\frac{L}{G}\right) e^{-NL/G} \\ &= e^{-NL/G}\left(1 - \frac{NL}{G}\right).\end{aligned}$$

Setting this equal to zero and noting $e^{-NL/G} > 0$:

$$1 - \frac{NL}{G} = 0 \implies N = \frac{G}{L}.$$

To confirm this is a maximum, we check the second derivative:

$$\begin{aligned}\frac{d^2}{dN^2}\left(Ne^{-NL/G}\right) &= \frac{d}{dN}\left[e^{-NL/G}\left(1 - \frac{NL}{G}\right)\right] \\ &= -\frac{L}{G}e^{-NL/G}\left(1 - \frac{NL}{G}\right) + e^{-NL/G}\left(-\frac{L}{G}\right) \\ &= e^{-NL/G}\left(-\frac{L}{G}\right)\left(2 - \frac{NL}{G}\right).\end{aligned}$$

Evaluating at $N = G/L$:

$$\left.\frac{d^2}{dN^2}\left(Ne^{-NL/G}\right)\right|_{N=G/L} = e^{-1}\left(-\frac{L}{G}\right)(2-1) = -\frac{L}{Ge} < 0,$$

since $L, G > 0$. The second derivative is negative, confirming that $N = G/L$ is a maximum.

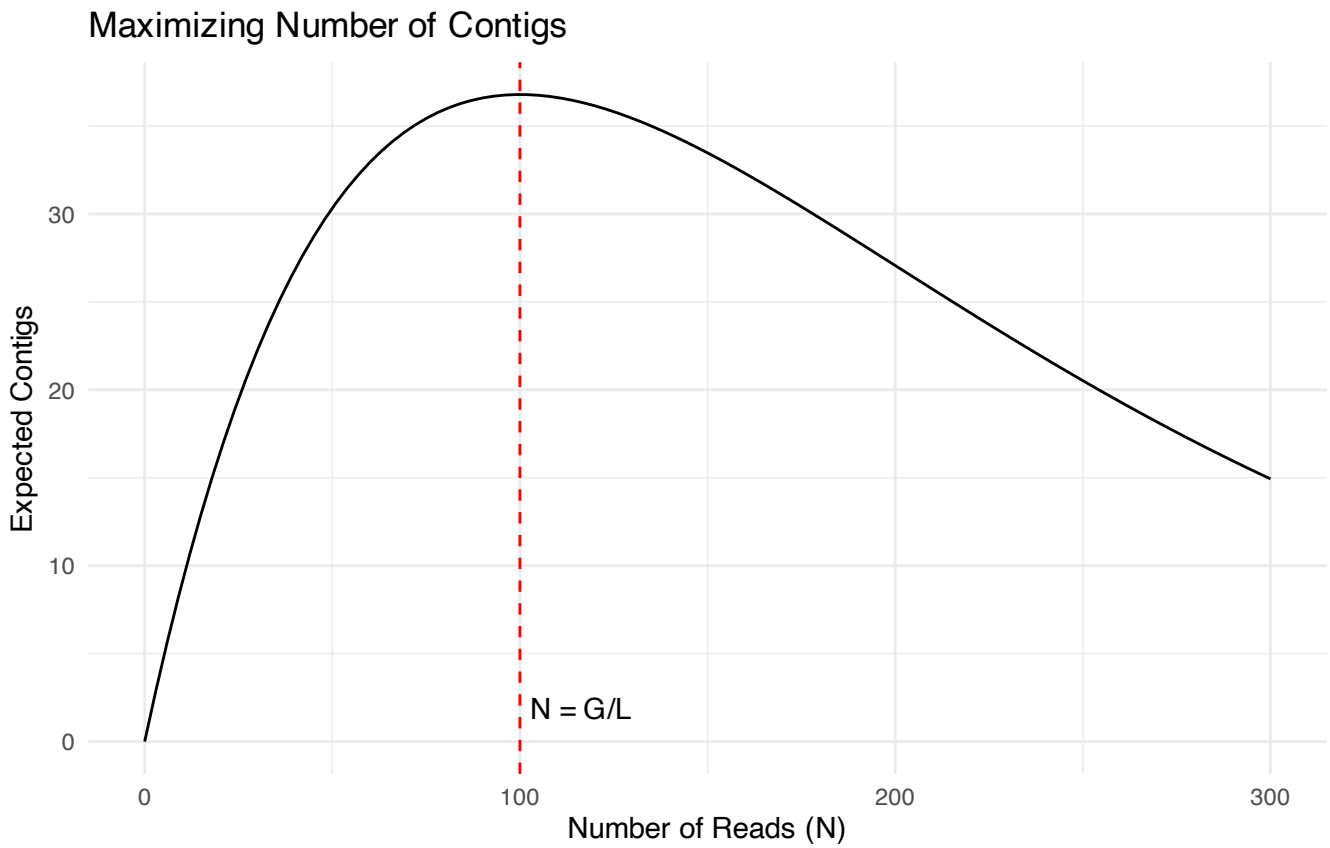

Figure 1: For $G = 100000$ and $L = 100$, we have the curve for expected contigs.

Bonus (not attempted)

6. **The following will walk you through approximating the variance of the mean contig size in a very simple case.**
    a. **Suppose that the length of each fragment is $L = 3$. Let C be the contig length beginning with some fragment F_1 with mean $\mu = E[C]$ and second moment $v_0 = E[C^2]$. Find expressions for $E[C^2|F_2]$, $E[C^2|F_2^c \cap F_3]$ and $E[C^2|F_2^c \cap F_3^c]$ in terms of μ and v_0. (Hint: The first step is $E[C^2|F_2] = E[(C+1)^2]$).**
    b. **Similar to how we solved for $E[C]$ in the notes, construct $E[C^2]$ using the law of total expectation (do this for a general $p = P(F_i)$ and $q = 1 - p$).**
    c. **Find an expression for $Var(C)$ (you can express your answer in terms of μ, but not v_0).**
    d. **For $L = 3$ and $a = 2, 4$ and 10, find the $E[C]$ and $Var(C)$.**

Appendix

```
knitr::opts_chunk$set(
  dev = "cairo_pdf",
  fig.width = 5,
  fig.height = 5,
  fig.align = "center",
  echo = FALSE,
  message = FALSE,
  warning = FALSE,
  error = FALSE,
  results = "asis"
)

# Load required libraries
library("tidyverse")
library("patchwork")
library("glue")
library("scales", warn.conflicts = FALSE)
library("extrafont")
library("tinytex")
library("knitr")
library("tidyr")
library("latex2exp")
library("MASS")
library("glue")
library("kableExtra")

theme_set(theme_minimal(base_family = "Roboto Condensed"))

conflicted::conflicts_prefer(
  readr::col_factor(),
  purrr::discard(),
  dplyr::lag(),
  readr::parse_date(),
  kableExtra::group_rows(),
  dplyr::select
)
# PROBLEM 2 ------------------------------------------------------------
compute_no_overlap <- function(k, N) {
  prob <- (1 / 4)k
  mu <- (N - k + 1) * prob
  sigsq <- (N - k + 1) *
    prob *
    (1 - prob) -
    2 * prob2 * (N - (3 / 2) * k + 1) * (k - 1)
```

```r
  list("mu" = mu, "sigsq" = sigsq)
}

get_overlaps <- function(word) {
  k <- nchar(word)
  shifts <- 1:(k - 1)

  valid_shifts <- keep(
    shifts,
    ~ {
      suffix <- str_sub(word, .x + 1, k)
      prefix <- str_sub(word, 1, k - .x)
      suffix == prefix
    }
  )

  valid_shifts
}

compute_with_overlap <- function(word, N) {
  k <- nchar(word)
  prob <- (1 / 4)k

  # Base variance from other function
  var_no_overlap <- compute_no_overlap(k, N)$sigsq

  overlaps <- get_overlaps(word)

  # Adjustment to account for overlaps
  # 2 * sum_{j in overlaps} (N - k + 1 - j) * P(overlap pattern)
  overlap_adj <- 0
  if (length(overlaps) > 0) {
    overlap_adj <- sum(map_dbl(
      overlaps,
      ~ {
        # The probability of the combined pattern of length k + j
        p_joint <- (1 / 4)(k + .x)
        2 * (N - k + 1 - .x) * p_joint
      }
    ))
  }

  list(
    "mu" = (N - k + 1) * prob,
    "sigsq" = var_no_overlap + overlap_adj
  )
}
```

```r
c(20, 100, 1000) |>
  map_dfr(
    ~ {
      res_no <- compute_no_overlap(5, .x)
      res_with <- compute_with_overlap("ACTAC", .x)
      tibble(
        N = .x,
        Mean = res_no$mu,
        Var_No = res_no$sigsq,
        Var_With = res_with$sigsq
      )
    }
  ) |>
  kable(
    booktabs = TRUE,
    digits = 6,
    col.names = c("N", "Mean", "Var (No Overlap)", "Var (With Overlap)")
  ) |>
  kable_styling(latex_options = "hold_position")

# PROBLEM 3 ----------------------------------------------------------
reverse_complement <- function(pattern) {
  comp <- chartr("ACGT", "TGCA", pattern)
  paste(rev(strsplit(comp, "")[[1]]), collapse = "")
}

FrequentWordsWithRC <- function(text, k) {
  n <- nchar(text)
  freqMap <- list()

  # Build frequency map
  for (i in 1:(n - k + 1)) {
    pattern <- substr(text, i, i + k - 1)
    rc <- reverse_complement(pattern)
    key <- min(pattern, rc)
    freqMap[[key]] <- (freqMap[[key]] %||% 0) + 1
  }

  counts <- unlist(freqMap)
  maxCount <- max(counts)

  names(counts[counts == maxCount])
}

`%||%` <- function(a, b) if (!is.null(a)) a else b

path <- "/Users/carson/Documents/Baylor_Work/2026_Spring/Statistical_Bioinfor
```

```r
dna_long <- read_lines(path) |>
  str_flatten()

result <- FrequentWordsWithRC(dna_long, 9)
cat("Most frequent k-mers (with RC grouping):", result[1:2], "\n", result[3:5]

# PROBLEM 5 ----------------------------------------------------------
# Parameters
G <- 10000 # Genome size
L <- 100 # Read length

expected_contigs <- function(N, G, L) {
  a <- (N * L) / G
  return(N * exp(-a))
}

peak_N <- G / L

ggplot(data.frame(N = 0:(3 * peak_N)), aes(x = N)) +
  geom_function(fun = expected_contigs, args = list(G = G, L = L)) +
  geom_vline(xintercept = peak_N, linetype = "dashed", color = "red") +
  annotate(
    "text",
    x = peak_N,
    y = 0,
    label = "N = G/L",
    vjust = -1,
    hjust = -0.1
  ) +
  labs(
    title = "Maximizing Number of Contigs",
    y = "Expected Contigs",
    x = "Number of Reads (N)"
  ) +
  theme_minimal()
```