Assignment 3

PH C240C

Fall 2016

Daniel Lee

Question 1

$$P(X=x)=\frac{n!}{\prod_{k=1}^{K} x_k!}\prod_{k=1}^{K}\pi_k^{x_k}$$

where $X=(X_k : k=1,\dots,K)$, a random variable.

$X \sim \text{Multinomial}(n, \pi=(\pi_k : k=1,\dots,K))$

$x=(x_k : k=1,\dots,K) \in \mathbb{N}^K$

$\sum_{k=1}^{K} x_k = n$

$\pi=(\pi_k : k=1,\dots,K) \in [0,1]^K$, with $\sum_{k=1}^{K}\pi_k = 1$

$$P(X_k=x_k \mid X_k+X_{k'}=x_k+x_{k'}) = \frac{P(X_k=x_k \cap (X_k+X_{k'}=x_k+x_{k'}))}{P(X_k+X_{k'}=x_k+x_{k'})}$$

$$= \frac{\binom{n}{x_k, x_{k'}, n-x_k-x_{k'}}\pi_k^{x_k}\pi_{k'}^{x_{k'}}(1-\pi_k-\pi_{k'})^{n-x_k-x_{k'}}}{\binom{n}{x_k+x_{k'}}(\pi_k+\pi_{k'})^{x_k+x_{k'}}(1-\pi_k-\pi_{k'})^{n-x_k-x_{k'}}}$$

Question 1 continued

$$= \frac{\frac{n!}{x_k!\,x_{k'}!\,(n-x_k-x_{k'})!}\,\pi_k^{x_k}\,\pi_{k'}^{x_{k'}}\left(1-\pi_k-\pi_{k'}\right)^{n-x_k-x_{k'}}}{\frac{n!}{(x_k+x_{k'})!\,(n-x_k-x_{k'})!}\left(\pi_k+\pi_{k'}\right)^{x_k+x_{k'}}\left(1-\pi_k-\pi_{k'}\right)^{n-x_k-x_{k'}}}$$

$$= \frac{\frac{1}{x_k!\,x_{k'}!}\,\pi_k^{x_k}\,\pi_{k'}^{x_{k'}}}{\frac{1}{(x_k+x_{k'})!}\left(\pi_k+\pi_{k'}\right)^{x_k+x_{k'}}}$$

$$= \frac{(x_k+x_{k'})!\;\pi_k^{x_k}\pi_{k'}^{x_{k'}}}{(x_k)!\,(x_{k'})!\,(\pi_k+\pi_{k'})^{x_k+x_{k'}}}$$

$$= \frac{(x_k+x_{k'})!}{(x_k)!\,(x_{k'})!}\left(\frac{\pi_k}{\pi_k+\pi_{k'}}\right)^{x_k}\left(\frac{\pi_{k'}}{\pi_k+\pi_{k'}}\right)^{x_{k'}}$$

$$\therefore\; X_k \mid X_k+X_{k'} \sim \text{Bin}\left(X_k+X_{k'},\, \frac{\pi_k}{\pi_k+\pi_{k'}}\right)$$

$$E\left[X_k \mid X_k+X_{k'}\right] = (X_k+X_{k'})\left(\frac{\pi_k}{\pi_k+\pi_{k'}}\right)$$

Question 3

Observed incomplete data structure:

Let $Y = (Y_A, Y_B, Y_{AB}, Y_O)$ denote the ABO phenotype counts for a random sample of n individuals from the population of interest.

Let $\pi = (\pi_A, \pi_B, \pi_O)$ denote the ABO allele frequencies in a well-defined population of interest.

Under the assumption of the Hardy-Weinberg equilibrium, the observed incomplete data structure is a multinomial distribution with four cells:

$Y \sim \text{Multinomial}(n, (\pi_A^2 + 2\pi_A\pi_O, \pi_B^2 + 2\pi_B\pi_O, 2\pi_A\pi_B, \pi_O^2))$

$$P(Y=y) = \frac{n!}{y_A! y_B! y_{AB}! y_O!} (\pi_A^2 + 2\pi_A\pi_O)^{y_A} (\pi_B^2 + 2\pi_B\pi_O)^{y_B} * (2\pi_A\pi_B)^{y_{AB}} (\pi_O^2)^{y_O}$$

$P(Y=y) = lik(\pi)$. That is, the above is the likelihood function for $\pi = (\pi_A, \pi_B, \pi_O)$ for the incomplete data.

$$\log lik(\pi) = \log n! - \log y_A! - \log y_B! - \log y_{AB}! - \log y_O! + y_A \log(\pi_A^2 + 2\pi_A\pi_O) + y_B \log(\pi_B^2 + 2\pi_B\pi_O) + y_{AB} \log(2\pi_A\pi_B) + 2 y_O \log(\pi_O)$$

where $\pi_A + \pi_B + \pi_O = 1$ and $Y_A + Y_B + Y_{AB} + Y_O = n$

Question 3 continued

Unobserved complete data structure:

Let $X = (X_{AA}, X_{AO}, X_{BB}, X_{BO}, X_{AB}, X_O)$ denote the ABO genotype counts for a random sample of n individuals from the population of interest.

Let $\pi = (\pi_A, \pi_B, \pi_O)$ denote the ABO allele frequencies in a well-defined population of interest.

The unobserved complete data structure is a multinomial distribution with six cells:

$$X \sim \text{Multinomial}(n, (\pi_A^2, 2\pi_A\pi_O, \pi_B^2, 2\pi_B\pi_O, 2\pi_A\pi_B, \pi_O^2))$$

$$P(X=x) = \frac{n!}{x_{AA}!\,x_{AO}!\,x_{BB}!\,x_{BO}!\,x_{AB}!\,x_O!} (\pi_A^2)^{x_{AA}} (2\pi_A\pi_O)^{x_{AO}} * (\pi_B^2)^{x_{BB}} (2\pi_B\pi_O)^{x_{BO}} (2\pi_A\pi_B)^{x_{AB}} (\pi_O^2)^{x_O}$$

where $\pi_A + \pi_B + \pi_O = 1$ and $x_{AA} + x_{AO} + x_{BB} + x_{BO} + x_{AB} + x_O = n$.

$$lik_c(\pi) = P(X=x)$$

$$\begin{aligned} \log lik_c(\pi) = {} & \log n! - \log x_A! - \log x_{AO}! - \log x_{BB}! - \log x_{BO}! \\ & - \log x_{AB}! - \log x_o! + 2x_{AA}\log\pi_A + x_{AO}\log(2\pi_A\pi_O) \\ & + 2x_{BB}\log(\pi_B) + x_{BO}\log(2\pi_B\pi_O) + x_{AB}\log(2\pi_A\pi_B) \\ & + 2x_O\log(\pi_O) \end{aligned}$$

Question 3 continued (E-Step)

$$Q(\Psi^{new}|\Psi^{old}) = E[\log lik_c(\Psi^{new}) \mid Y=y; \Psi^{old}]$$

$$Q(\pi^{new}|\pi^{old}) = E[\log lik_c(\pi^{new}) \mid Y=y; \pi^{old}]$$

$$= E\Big[\log n! - \log x_{AA}! - \log x_{AO}! - \log x_{BB}! - \log x_{BO}! - \log x_{AB}!$$
$$- \log x_O! + 2X_{AA}\log \pi_A^{new} + X_{AO}\log(2\pi_A^{new}\pi_O^{new}) +$$
$$2X_{BB}\log(\pi_B^{new}) + X_{BO}\log(2\pi_B^{new}\pi_O^{new})$$
$$+ X_{AB}\log(2\pi_A^{new}\pi_B^{new}) + 2X_O\log(\pi_O^{new}) \mid Y=y; \pi^{old}\Big]$$

$$= \log n! - \log x_{AA}! - \log x_{AO}! - \log x_{BB}! - \log x_{BO}!$$
$$- \log x_{AB}! - \log x_O! + 2\log \pi_A^{new} E[X_{AA} \mid Y=y; \pi^{old}]$$
$$+ \log(2\pi_A^{new}\pi_O^{new}) E[X_{AO} \mid Y=y; \pi^{old}]$$
$$+ 2\log(\pi_B^{new}) E[X_{BB} \mid Y=y; \pi^{old}]$$
$$+ \log(2\pi_B^{new}\pi_O^{new}) E[X_{BO} \mid Y=y; \pi^{old}]$$
$$+ \log(2\pi_A^{new}\pi_B^{new}) E[X_{AB} \mid Y=y; \pi^{old}]$$
$$+ 2\log(\pi_O^{new}) E[X_O \mid Y=y; \pi^{old}]$$

Since $X_{AA} + X_{AO} = Y_A$, $X_{BB} + X_{BO} = Y_B$,

$X_{AB} = Y_{AB}$, $X_O = Y_O$, we have the following:

Question 3 continued (E-Step)

$Q(\pi^{new}/\pi^{old})$

$= \log n! - \log x_{AA}! - \log x_{AO}! - \log x_{BB}! - \log x_{BO}! - \log x_{AB}! - \log x_O!$

$+ 2\log \pi_A^{new} E[X_{AA}|Y_A;\pi^{old}]$

$+ \log(2\pi_A^{new}\pi_O^{new}) E[X_{AO}|Y_A;\pi^{old}]$

$+ 2\log(\pi_B^{new}) E[X_{BB}|Y_B;\pi^{old}]$

$+ \log(2\pi_B^{new}\pi_O^{new}) E[X_{BO}|Y_B;\pi^{old}]$

$+ \log(2\pi_A^{new}\pi_B^{new}) E[X_{AB}|Y_{AB};\pi^{old}]$

$+ 2\log(\pi_O^{new}) E[X_O|Y_O;\pi^{old}]$

From question 1, we know that

$$X_{AA}|Y_A;\pi^{old} \sim Bin\left(Y_A, \frac{\pi_A^{old\,2}}{\pi_A^{old\,2} + 2\pi_A^{old}\pi_O^{old}}\right)$$

$$X_{AO}|Y_A;\pi^{old} \sim Bin\left(Y_A, \frac{2\pi_A^{old}\pi_O^{old}}{\pi_A^{old\,2} + 2\pi_A^{old}\pi_O^{old}}\right)$$

$$X_{BB}|Y_B;\pi^{old} \sim Bin\left(Y_B, \frac{\pi_B^{old\,2}}{\pi_B^{old\,2} + 2\pi_B^{old}\pi_O^{old}}\right)$$

$$X_{BO}|Y_B;\pi^{old} \sim Bin\left(Y_B, \frac{2\pi_B^{old}\pi_O^{old}}{\pi_B^{old\,2} + 2\pi_B^{old}\pi_O^{old}}\right)$$

Question 3 continued (E-Step)

Also, $E[X_{AB} | Y_{AB}; \pi^{old}] = Y_{AB}$

$E[X_0 | X_0; \pi^{old}] = Y_0.$

So we have

$Q(\pi^{new} | \pi^{old})$

$$= \log n! - \log x_{AA}! - \log x_{A0}! - \log x_{BB}! - \log x_{B0}!$$
$$- \log x_{AB}! - \log x_0!$$
$$+ 2\log(\pi_A^{new})(Y_A)\left(\frac{\pi_A^{old2}}{\pi_A^{old2} + 2\pi_A^{old}\pi_0^{old}}\right)$$
$$+ \log(2\pi_A^{new}\pi_0^{new})(Y_A)\left(\frac{2\pi_A^{old}\pi_0^{old}}{\pi_A^{old2} + 2\pi_A^{old}\pi_0^{old}}\right)$$
$$+ 2\log(\pi_B^{new})(Y_B)\left(\frac{\pi_B^{old2}}{\pi_B^{old2} + 2\pi_B^{old}\pi_0^{old}}\right)$$
$$+ \log(2\pi_B^{new}\pi_0^{new})(Y_B)\left(\frac{2\pi_B^{old}\pi_0^{old}}{\pi_B^{old2} + 2\pi_B^{old}\pi_0^{old}}\right)$$
$$+ \log(2\pi_A^{new}\pi_B^{new})\, Y_{AB}$$
$$+ 2\log(\pi_0^{new})\, Y_0$$

Question 3 continued (M-Step)

M-Step:

Since I have a constraint that $\pi_A + \pi_B + \pi_O = 1$,

I will introduce a Lagrange multiplier and maximize

$$Q(\pi^{new} | \pi^{old}) = E[\log lik_c(\pi^{new}) | Y = y; \pi^{old}].$$

$$E[L_c(\pi^{new}) | Y = y; \pi^{old}]$$

$$= E[\log lik_c(\pi^{new}) | Y = y; \pi^{old}] + \lambda(\pi_A + \pi_B + \pi_O - 1)$$

$$\frac{\partial E[L_c(\pi^{new}) | Y = y; \pi^{old}]}{\partial \pi_A^{new}} = 0$$

$$= \frac{2Y_A}{\pi_A^{new}} \left(\frac{\pi_A^{old\,2}}{\pi_A^{old\,2} + 2\pi_A^{old}\pi_O^{old}} \right) + \frac{Y_A}{\pi_A^{new}} \left(\frac{2\pi_A^{old}\pi_O^{old}}{\pi_A^{old\,2} + 2\pi_A^{old}\pi_O^{old}} \right)$$

$$+ \frac{Y_{AB}}{\pi_A^{new}} + \lambda$$

$$\pi_A^{new} = \frac{2Y_A}{-\lambda} \left(\frac{\pi_A^{old\,2}}{\pi_A^{old\,2} + 2\pi_A^{old}\pi_O^{old}} \right) + \frac{Y_A}{-\lambda} \left(\frac{2\pi_A^{old}\pi_O^{old}}{\pi_A^{old\,2} + 2\pi_A^{old}\pi_O^{old}} \right)$$

$$+ \frac{Y_{AB}}{-\lambda}$$

Question 3 continued (M-Step)

$$\frac{\partial E[L_c(\pi^{new}) \mid Y=y; \pi^{old}]}{\partial \pi_B^{new}} = 0$$

$$= \frac{2Y_B}{\pi_B^{new}}\left(\frac{\pi_B^{old\,2}}{\pi_B^{old\,2} + 2\pi_B^{old}\pi_0^{old}}\right) + \frac{Y_B}{\pi_B^{new}}\left(\frac{2\pi_B^{old}\pi_0^{old}}{\pi_B^{old\,2} + 2\pi_B^{old}\pi_0^{old}}\right)$$

$$+ \frac{Y_{AB}}{\pi_B^{new}} + \lambda$$

$$\pi_B^{new} = \frac{2Y_B}{-\lambda}\left(\frac{\pi_B^{old\,2}}{\pi_B^{old\,2} + 2\pi_B^{old}\pi_0^{old}}\right) + \frac{Y_B}{-\lambda}\left(\frac{2\pi_B^{old}\pi_0^{old}}{\pi_B^{old\,2} + 2\pi_B^{old}\pi_0^{old}}\right)$$

$$+ \frac{Y_{AB}}{-\lambda}$$

$$\frac{\partial E[L_c(\pi^{new}) \mid Y=y; \pi^{old}]}{\partial \pi_0^{new}} = 0$$

$$= \frac{Y_A}{\pi_0^{new}}\left(\frac{2\pi_A^{old}\pi_0^{old}}{\pi_A^{old\,2} + 2\pi_A^{old}\pi_0^{old}}\right) + \frac{Y_B}{\pi_0^{new}}\left(\frac{2\pi_B^{old}\pi_0^{old}}{2\pi_B^{old\,2} + 2\pi_B^{old}\pi_0^{old}}\right)$$

$$+ \frac{2Y_0}{\pi_0^{new}}$$

Question 3 continued (M-Step)

$$\pi_0^{new} = \frac{Y_A}{-\lambda}\left(\frac{2\pi_A^{old}\pi_0^{old}}{\pi_A^{old2}+2\pi_A^{old}\pi_0^{old}}\right) + \frac{Y_B}{-\lambda}\left(\frac{2\pi_B^{old}\pi_0^{old}}{\pi_B^{old2}+2\pi_B^{old}\pi_0^{old}}\right)$$

$$+ \frac{2Y_0}{-\lambda}$$

Sum the three equations:

$$(\pi_A^{new} + \pi_B^{new} + \pi_0^{new})(-\lambda)$$

$$= 2Y_A\left(\frac{\pi_A^{old2}}{\pi_A^{old2}+2\pi_A^{old}\pi_0^{old}}\right) + 4Y_A\left(\frac{\pi_A^{old}\pi_0^{old}}{\pi_A^{old2}+2\pi_A^{old}\pi_0^{old}}\right)$$

$$+ 2Y_{AB} + 2Y_0$$

$$+ 2Y_B\left(\frac{\pi_B^{old2}}{\pi_B^{old2}+2\pi_B^{old}\pi_0^{old}}\right) + 4Y_B\left(\frac{\pi_B^{old}\pi_0^{old}}{\pi_B^{old2}+2\pi_B^{old}\pi_0^{old}}\right)$$

$$-\lambda = 2Y_A\left(\frac{\pi_A^{old2}+2\pi_A^{old}\pi_0^{old}}{\pi_A^{old2}+2\pi_A^{old}\pi_0^{old}}\right) + 2Y_{AB} + 2Y_0$$

$$+ 2Y_B\left(\frac{\pi_B^{old2}+2\pi_B^{old}\pi_0^{old}}{\pi_B^{old2}+2\pi_B^{old}\pi_0^{old}}\right)$$

$$-\lambda = 2n$$

Question 3 continued (M-step)

Plugging in $-\lambda = 2n$:

$$\pi_A^{new} = \frac{2Y_A}{2n}\left(\frac{\pi_A^{old2}}{\pi_A^{old2} + 2\pi_A^{old}\pi_0^{old}}\right) + \frac{Y_A}{2n}\left(\frac{2\pi_A^{old}\pi_0^{old}}{\pi_A^{old2} + 2\pi_A^{old}\pi_0^{old}}\right)$$

$$+ \frac{Y_{AB}}{2n}$$

$$\pi_B^{new} = \frac{2Y_B}{2n}\left(\frac{\pi_B^{old2}}{\pi_B^{old2} + 2\pi_B^{old}\pi_0^{old}}\right) + \frac{Y_B}{2n}\left(\frac{2\pi_B^{old}\pi_0^{old}}{\pi_B^{old2} + 2\pi_B^{old}\pi_0^{old}}\right)$$

$$+ \frac{Y_{AB}}{2n}$$

$$\pi_0^{new} = \frac{Y_A}{2n}\left(\frac{2\pi_A^{old}\pi_0^{old}}{\pi_A^{old2} + 2\pi_A^{old}\pi_0^{old}}\right) + \frac{Y_B}{2n}\left(\frac{2\pi_B^{old}\pi_0^{old}}{\pi_B^{old2} + 2\pi_B^{old}\pi_0^{old}}\right)$$

$$+ \frac{2Y_0}{2n}$$

Assignment 3

Daniel Lee

November 22, 2016

Question 2

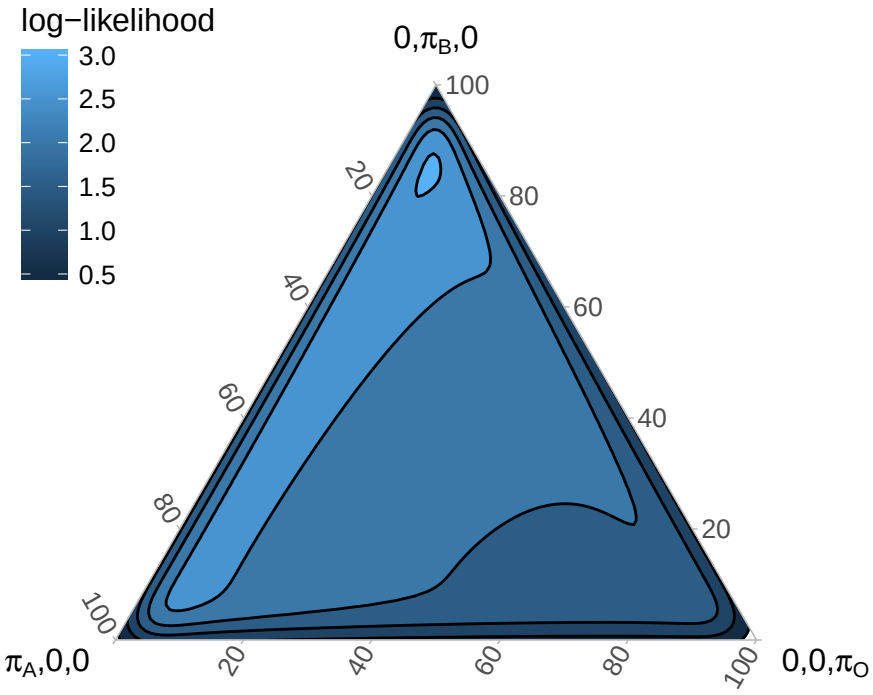
Log-likelihood Surface for Trinomial Probabilities

```
require(devtools)
require(grid)
require(ggtern)
require(geometry)
require(RColorBrewer)
require(reshape2)
```

```
myPaletteSeq <- brewer.pal(9,'Set1')

# Create a matrix with all possible combinations of pi_a and pi_b
all_pi_combinations <- expand.grid(pi_a = seq(0, 1, by = 0.01),
                         pi_b = seq(0, 1, by = 0.01))

# Select pi_a and pi_b values greater than zero
all_pi_combinations <- all_pi_combinations[all_pi_combinations$pi_a > 0
                                           & all_pi_combinations$pi_b > 0,]

# Select rows where pi_a + pi_b is less than 1
# Do not select rows where pi_a + pi_b = 1 because then pi_o = 0, which
# would cause an error because log(0) = negative infinity
all_pi_combinations <- all_pi_combinations[(all_pi_combinations$pi_a +
                                              all_pi_combinations$pi_b < 1),]

# Add pi_o column
all_pi_combinations$pi_o <- 1 - (all_pi_combinations$pi_a + all_pi_combinations$pi_b)

# Data from the Clark dataset
Y_A <- 186
Y_B <- 38
Y_AB <- 13
Y_O <- 284

# Initial values of pi_a, pi_b, pi_o
pi <- c(1/3, 1/3, 1/3)

#Description: function that calculates log-likelihood of incomplete
# data structure
# Input
# pi_a: allele frequency pi_a
# pi_b: allele frequency pi_b
# pi_o: allele frequency pi_o
```

```r
# Output
# loglikelihood: numeric
loglik <- function(Y_A, Y_B, Y_AB, Y_O, pi){

  pi_a <- pi[1]
  pi_b <- pi[2]
  pi_o <- pi[3]
  n <- pi_a + pi_b + pi_o

  #log-likelihood for incomplete data structure
  loglikelihood <- lgamma(n + 1) -
    lgamma(Y_A + 1) -
    lgamma(Y_B + 1) -
    lgamma(Y_O + 1) -
    lgamma(Y_AB + 1) + 2 * Y_O * log(pi_o) +
    Y_A * log(pi_a ^ 2 + 2 * pi_a * pi_o) +
    Y_B * log(pi_b ^ 2 + 2 * pi_b * pi_o) +
    Y_AB * log(2 * pi_a * pi_b)

  return(loglikelihood)

}

# Calculate the log-likelihood for every combination of pi
all_loglik <- apply(all_pi_combinations, 1, function(x) {
  loglik(Y_A, Y_B, Y_AB, Y_O, x)})


all_pi_combinations$loglik <- all_loglik

# Barycentric Coordinates
ggtern(data = all_pi_combinations,
       (aes(x = pi_a, y = pi_b, z = pi_o))) +
  stat_density_tern(geom = "polygon", color = "black",
                    n = 500, h = 0.75, expand = 1.5,
                    base = 'identity',
                    aes(fill = ..level.., weight = loglik),
                    na.rm = TRUE) +
    theme_light() +
    theme(legend.position = c(0,1),
          legend.justification = c(0,1)) +
    labs(fill = "log-likelihood",
          colour = "log-likelihood",
          title = "Log-likelihood Surface for Observed Incomplete Data Structure") +
    Llab(expression(paste(pi[A],",0,0"))) +
    Tlab(expression(paste("0,",pi[B],",0"))) +
    Rlab(expression(paste("0,0,",pi[O])))
```

Log−likelihood Surface for Observed Incomplete Data Structure



Question 4

Software Implementation of EM Algorithm

```
# DESCRIPTION: This function implenets the EM algorithm for MLE estimation based on input data.
# Then, this function creates an object of class "EM_ABO" that contains
# information of the EM algorithm implementation.
# of the ABO allele frequencies
# INPUT
# Y_A: observed phenotype counts of blood group A
# Y_B: observed phenotype counts of blood group B
# Y_O: observed phenotype counts of blood group O
# Y_AB: observed phenotype counts of blood group AB
# pi_a_old: starting value for pi_a allele frequency
# pi_b_old: starting value for pi_b allele frequency
# pi_o_old: starting value for pi_o allele frequency
# stop_criteria: stopping criteria for EM algorithm iteration based on
# difference in absolute value of log-likelihoods from k-1 and k iterations
# OUTPUT
# Calls the print method to print candidate MLEs for the allele
# frequencies along with observed data log-likelihood
# res: list that contains last iteration number, MLEs for the allele frequencies,
# final observed data log-likelihood, and a data frame that logs all the
# iterations of the EM algorithm (not printed but contained in the object)
EM_ABO <- function(Y_A,
                   Y_B,
                   Y_O,
                   Y_AB,
                   pi_a_old,
                   pi_b_old,
                   pi_o_old,
                   stop_criteria){

  # Description: function that calculates allele frequencies for new iteration
  # using allele frequencies from previous iteration
  # Input
  # pi_a: old allele frequency pi_a
  # pi_b: old allele frequency pi_b
  # pi_o: old allele frequency pi_o
  # Output
  # a vector of new allele frequencies
  update_pi <- function(pi_a, pi_b, pi_o){

    pi_a_new <- (2 * Y_A * (pi_a ^ 2 / (pi_a ^ 2 + 2 * pi_a * pi_o)) +
                 Y_A * (2 * pi_a * pi_o / (pi_a ^ 2 + 2 * pi_a * pi_o)) +
                  Y_AB) / (2 * n)

    pi_b_new <- (2 * Y_B * (pi_b ^ 2 / (pi_b ^ 2 + 2 * pi_b * pi_o)) +
                 Y_B * (2 * pi_b * pi_o / (pi_b ^ 2 + 2 * pi_b * pi_o)) +
                  Y_AB) / (2 * n)

    pi_o_new <-  (Y_A * (2 * pi_a * pi_o / (pi_a ^ 2 + 2 * pi_a * pi_o)) +
                 Y_B * (2 * pi_b * pi_o / (pi_b ^ 2 + 2 * pi_b * pi_o)) +
```

```
              2 * Y_O) / (2 * n)

  return(c(pi_a_new, pi_b_new, pi_o_new))

}

# Description: function that calculates log-likelihood of incomplete
# data structure
# Input
# pi_a: allele frequency pi_a
# pi_b: allele frequency pi_b
# pi_o: allele frequency pi_o
# Output
# loglikelihood: numeric
loglik <- function(pi_a, pi_b, pi_o){

  #log-likelihood for incomplete data structure
  loglikelihood <- lgamma(n + 1) -
    lgamma(Y_A + 1) -
    lgamma(Y_B + 1) -
    lgamma(Y_O + 1) -
    lgamma(Y_AB + 1) + 2 * Y_O * log(pi_o) +
    Y_A * log(pi_a ^ 2 + 2 * pi_a * pi_o) +
    Y_B * log(pi_b ^ 2 + 2 * pi_b * pi_o) +
    Y_AB * log(2 * pi_a * pi_b)

  return(loglikelihood)

}

# n: total sample size
n <- Y_A  + Y_B + Y_AB + Y_O

# k: number of iterations in the EM algorithm
k <- 0

# log likelihood for iteration k = 0
log_lik_old <- loglik(pi_a_old, pi_b_old, pi_o_old)

# Initiate matrix of EM algorithm iterations
# Contents to this matrix will be updated with new iterations
iterations_table <- matrix(c(k, pi_a_old, pi_b_old, pi_o_old, log_lik_old),
                           nrow = 1,
                           ncol = 5,
                         dimnames =
                           list(c(), c("Iteration", "pi_a", "pi_b", "pi_o", "log_lik")))

# Update number of iterations k
k <- 1

# Allele frequency estimates for 1st iteration of EM algorithm
pi_new <- update_pi(pi_a_old, pi_b_old, pi_o_old)
pi_a_new <- pi_new[1]
```

```
  pi_b_new <- pi_new[2]
  pi_o_new <- pi_new[3]

  # log likelihood for iteration k = 1
  log_lik_new <- loglik(pi_a_new, pi_b_new, pi_o_new)

  # Update EM algorithm iterations matrix with 1st iteration added
  iterations_table <- rbind(iterations_table, c(k, pi_new, log_lik_new))

  # Perform iterations of EM algorithm until stop criteria is met
  while(abs(log_lik_new - log_lik_old) > stop_criteria){

    # Update iteration number k
    k <- k + 1

    # Calculate new estimates of allele frequencies
    pi_new <- update_pi(pi_a_new, pi_b_new, pi_o_new)
    pi_a_new <- pi_new[1]
    pi_b_new <- pi_new[2]
    pi_o_new <- pi_new[3]

    # Calculate updated log likelihood value using new allele frequencies
    log_lik_old <- log_lik_new
    log_lik_new <- loglik(pi_a_new, pi_b_new, pi_o_new)
    iterations_table <- rbind(iterations_table, c(k, pi_new, log_lik_new))

  }

  # List of objects for the "EM_ABO" object
  res <- list(
    iteration = k,
    pi_a_mle = pi_a_new,
    pi_b_mle = pi_b_new,
    pi_o_mle = pi_o_new,
    log_likelihood = log_lik_new,
    iterations_table = as.data.frame(iterations_table)
  )

  # Define class to access the print function for "EM_ABO" class
  class(res) <- "EM_ABO"
  res

}

# Description: prints summary information for 'EM_ABO' object
# Input
# x: object of class 'EM_ABO'
# Output
# the allele frequences mle and log likelihood value for the last iteration
print.EM_ABO <- function(x) {

  cat('pi_a_mle:', x$pi_a_mle, '\n')
  cat('pi_b_mle:', x$pi_b_mle, '\n')
```

```
  cat('pi_o_mle:', x$pi_o_mle, '\n')
  cat('log_likelihood:', x$log_likelihood, '\n')
  invisible(x)

}
```

Question 5

Application of EM Algorithm to Clark Dataset Using Initial Values of $\pi_A = 1/3$, $\pi_B = 1/3$, and $\pi_O = 1/3$

```
# data from the Clark dataset
Y_A <- 186

Y_B <- 38

Y_AB <- 13

Y_O <- 284

n <- Y_A + Y_B + Y_AB + Y_O

stop_criteria <- 0.00000001

#initial values of the allele frequencies to pass to EM algorithm
pi_a_old <- 1/3

pi_b_old <- 1/3

pi_o_old <- 1/3

MLE_Clark_data <- EM_ABO(Y_A, Y_B, Y_O, Y_AB, pi_a_old, pi_b_old, pi_o_old, stop_criteria)

# candidate MLEs for the allele frequencies
# and corresponding value of the observed data log-likelihood
# for final estimate
MLE_Clark_data
```

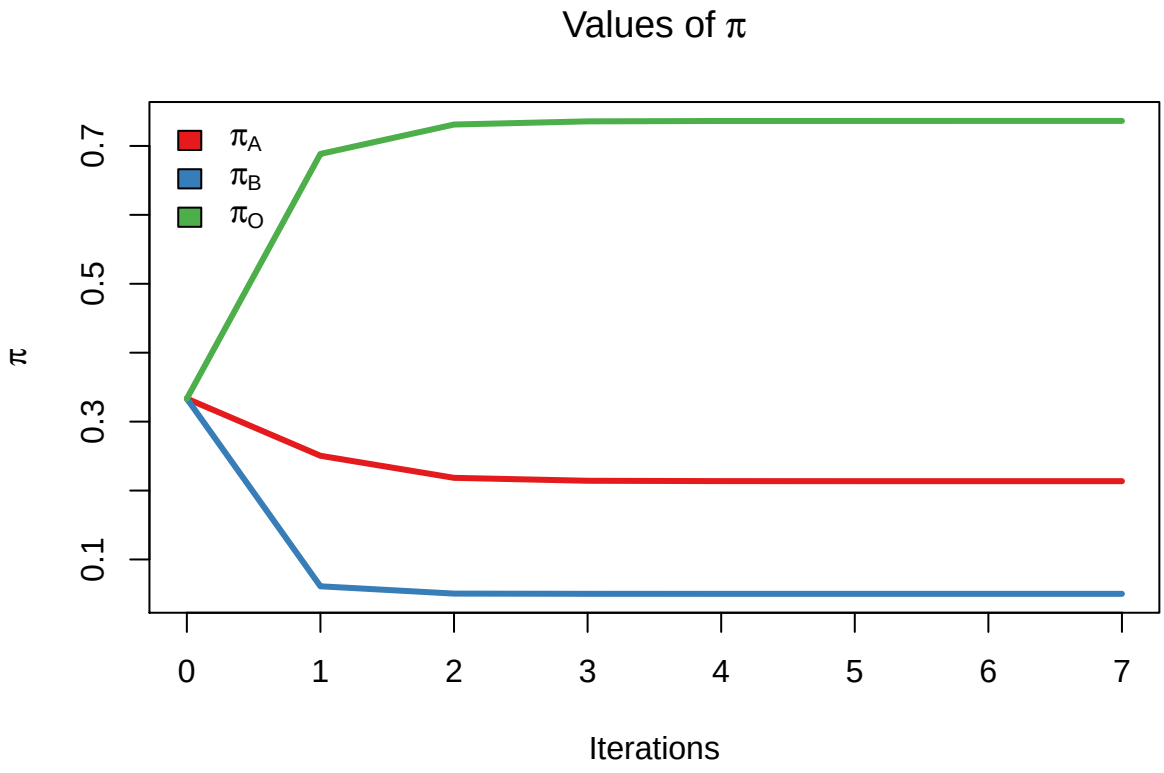
```
## pi_a_mle: 0.2135911
## pi_b_mle: 0.05014533
## pi_o_mle: 0.7362636
## log_likelihood: -8.372631
```

```
# Progress of EM algorithm for Clark data for every iteration
iterations_table <- MLE_Clark_data$iterations_table
iterations_table
```

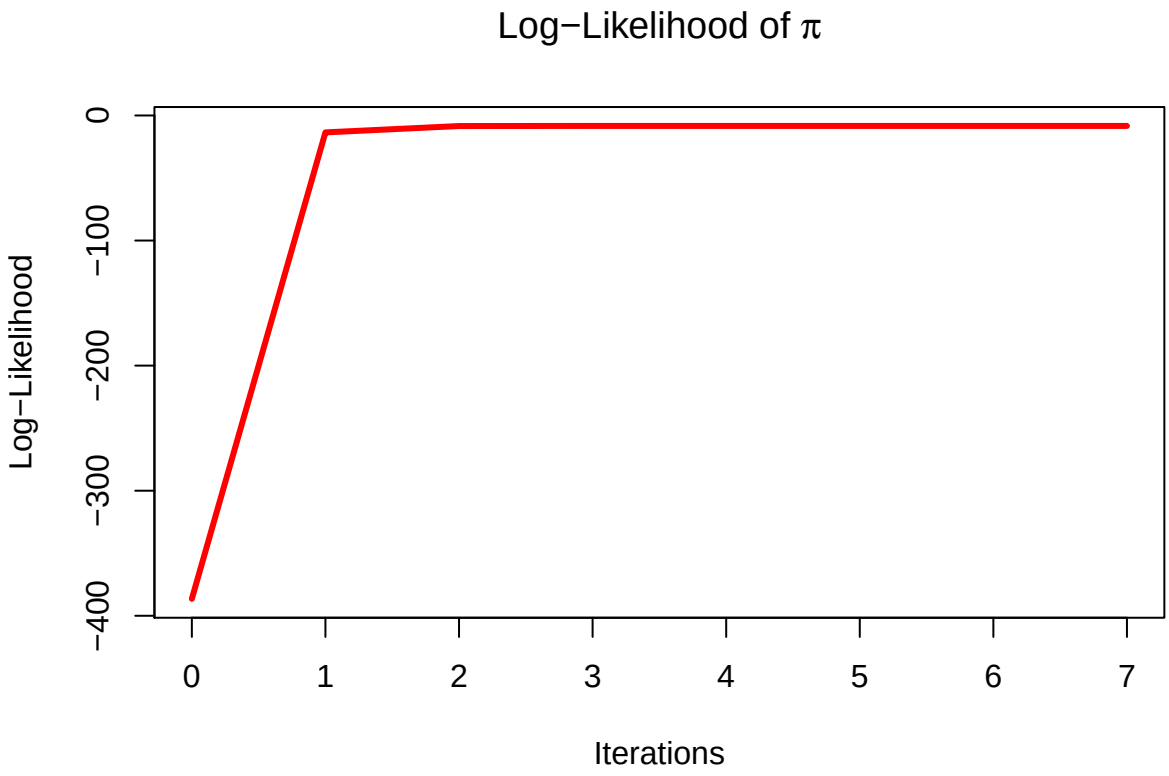
```
##   Iteration      pi_a       pi_b      pi_o     log_lik
## 1         0 0.3333333 0.33333333 0.3333333 -386.455100
```

```
## 2         1 0.2504798 0.06110045 0.6884197  -13.533052
## 3         2 0.2184544 0.05049394 0.7310517   -8.440824
## 4         3 0.2141823 0.05016173 0.7356559   -8.373613
## 5         4 0.2136619 0.05014667 0.7361914   -8.372645
## 6         5 0.2135994 0.05014547 0.7362551   -8.372631
## 7         6 0.2135920 0.05014535 0.7362627   -8.372631
## 8         7 0.2135911 0.05014533 0.7362636   -8.372631
```

```r
labels <- c(expression(pi[A]),
  expression(pi[B]),
  expression(pi[O]))

# plot of of the change in pi values as iterations change in the EM algorithm
matplot(iterations_table$Iteration,
        iterations_table[,2:4],
        type = "l",
        xlab = "Iterations",
        ylab = expression(pi),
        lty = 1, lwd = 3,
        main = expression(paste("Values of ", pi)),
        col = myPaletteSeq)
legend("topleft", labels,
       col=myPaletteSeq,
       fill=myPaletteSeq,
       bty = "n")
```



```r
# Log-likelihood of pi for incomplete data
plot(iterations_table$Iteration,
     iterations_table$log_lik,
     type = "l",
      xlab = "Iterations",
      ylab = "Log-Likelihood",
      lty = 1, lwd = 3,
      main = expression(paste("Log-Likelihood of ", pi)),
      col = "red")
```



Application of EM Algorithm to Clark Dataset Using Different Initial Values of the Allele Frequencies

```r
#1
pi_a_old <- 1
pi_b_old <- 0
pi_o_old <- 0

# The commented code below causes an error because the log-likelihood
# function becomes negative infinity
#EM_1 <- EM_ABO(Y_A, Y_B, Y_O, Y_AB, pi_a_old, pi_b_old, pi_o_old, stop_criteria)

#2
```

```r
pi_a_old <- .9999999
pi_b_old <- (1 - pi_a_old) / 2
pi_o_old <- pi_b_old

EM_2 <- EM_ABO(Y_A, Y_B, Y_O, Y_AB, pi_a_old, pi_b_old, pi_o_old, stop_criteria)

EM_2
```

```
## pi_a_mle: 0.213591
## pi_b_mle: 0.05014533
## pi_o_mle: 0.7362637
## log_likelihood: -8.372631
```

```r
iterations_table <- EM_2$iterations_table
iterations_table
```

```
##   Iteration      pi_a       pi_b       pi_o       log_lik
## 1         0 0.9999999 0.00000005 0.00000005 -10491.029541
## 2         1 0.3694817 0.06110045 0.56941780    -64.975209
## 3         2 0.2347055 0.05080130 0.71449324     -9.574897
## 4         3 0.2161612 0.05019630 0.73364251     -8.391014
## 5         4 0.2138994 0.05015066 0.73594997     -8.372897
## 6         5 0.2136279 0.05014595 0.73622617     -8.372635
## 7         6 0.2135954 0.05014540 0.73625924     -8.372631
## 8         7 0.2135915 0.05014534 0.73626319     -8.372631
## 9         8 0.2135910 0.05014533 0.73626367     -8.372631
```

```r
#3
pi_b_old <- .9999999
pi_a_old <- (1 - pi_b_old) / 2
pi_o_old <- pi_a_old

EM_3 <- EM_ABO(Y_A, Y_B, Y_O, Y_AB, pi_a_old, pi_b_old, pi_o_old, stop_criteria)

EM_3
```

```
## pi_a_mle: 0.2135911
## pi_b_mle: 0.05014533
## pi_o_mle: 0.7362636
## log_likelihood: -8.372631
```

```r
iterations_table <- EM_3$iterations_table
iterations_table
```

```
##   Iteration       pi_a       pi_b       pi_o       log_lik
## 1         0 0.00000005 0.99999990 0.00000005 -15304.562786
## 2         1 0.25047985 0.08541266 0.66410749    -23.161976
## 3         2 0.21930062 0.05114779 0.72955159     -8.479520
## 4         3 0.21430211 0.05017941 0.73551847     -8.374072
## 5         4 0.21367672 0.05014730 0.73617598     -8.372651
## 6         5 0.21360122 0.05014551 0.73625326     -8.372631
## 7         6 0.21359217 0.05014535 0.73626248     -8.372631
## 8         7 0.21359109 0.05014533 0.73626358     -8.372631
```

```
#4
pi_o_old <- .9999999
pi_a_old <- (1 - pi_o_old) / 2
pi_b_old <- pi_a_old

EM_4 <- EM_ABO(Y_A, Y_B, Y_O, Y_AB, pi_a_old, pi_b_old, pi_o_old, stop_criteria)

EM_4
```

```
## pi_a_mle: 0.2135909
## pi_b_mle: 0.05014533
## pi_o_mle: 0.7362638
## log_likelihood: -8.372631
```

```
iterations_table <- EM_4$iterations_table
iterations_table
```

```
##   Iteration       pi_a       pi_b      pi_o      log_lik
## 1         0 0.00000005 0.00000005 0.9999999 -3535.336061
## 2         1 0.19097889 0.04894434 0.7600768    -9.916601
## 3         2 0.21090153 0.05008188 0.7390166    -8.393042
## 4         3 0.21326899 0.05013954 0.7365915    -8.372921
## 5         4 0.21355239 0.05014468 0.7363029    -8.372635
## 6         5 0.21358632 0.05014525 0.7362684    -8.372631
## 7         6 0.21359039 0.05014532 0.7362643    -8.372631
## 8         7 0.21359087 0.05014533 0.7362638    -8.372631
```

When the initial values of the allele frequencies is zero for any of π_A, π_B, or π_O, the EM algorithm does not work because the log-likelihood becomes negative infinity. Other than that, it seems as though the EM algorithm converges to the same estimates of the allele frequencies regardless of the initial values of the allele frequencies. The convergence seems to be pretty quick. At most, it takes nine iterations to converge to an estimate.

Compare Results of EM Algorithm to One of the Optimization Functions In R

I will use the `solnp()` function from the package `Rsolnp`, which performs optimizations for functions with linear equality constraints.

```
require(Rsolnp)
```

```
## Loading required package: Rsolnp
```

```
## Warning: package 'Rsolnp' was built under R version 3.3.2
```

```
# Negative log-likelihood to minimize
# This is the negative log-likelihood for the observed incomplete data structure
negloglik <- function(par){

    pi_a <- par[1]
    pi_b <- par[2]
```

```
    pi_o <- par[3]

    #log-likelihood
    negloglikelihood <- -(lgamma(n + 1) -
      lgamma(Y_A + 1) -
      lgamma(Y_B + 1) -
      lgamma(Y_O + 1) -
      lgamma(Y_AB + 1) + 2 * Y_O * log(pi_o) +
      Y_A * log(pi_a ^ 2 + 2 * pi_a * pi_o) +
      Y_B * log(pi_b ^ 2 + 2 * pi_b * pi_o) +
      Y_AB * log(2 * pi_a * pi_b))

    return(negloglikelihood)

}

# Linear equality constraint
# pi_a + pi_b + pi_o = 1
eqn1 <- function(x){
  z1 = x[1] + x[2] + x[3]
  return(z1)
}
```

```
# Optimize the negative log-likelihood function
optimized <- solnp(par = c(pi_a_old, pi_b_old, pi_o_old),
      fun = negloglik, eqfun = eqn1, eqB = 1)
```

```
# Optimized estimates from solnp function
optimized$pars
```

```
## [1] 0.21359094 0.05014533 0.73626373
```

```
# MLE estimate from EM Algorithm implementation
MLE_Clark_data
```

```
## pi_a_mle: 0.2135911
## pi_b_mle: 0.05014533
## pi_o_mle: 0.7362636
## log_likelihood: -8.372631
```

The estimates from the implementation of the EM algorithm and the optimized values from `solnp()` function are very similar.