BIOINF 702 Assignment 1

Badi James (bjam575)

August 20, 2018

1 Introduction

Blood type is a phenotype that is determined by one genetic loci and is inherited in a mendelian fashion. For loci where there are only two allele types that either have a dominant recessive relationship or a distinct third phenotype occurring in heterozygous, if the population is in Hardy-Weinberg equilibrium then allele frequencies can be derived from phenotype frequencies. For alleles D and d with frequencies p_D and p_d this can be found using the HW equations: $p_{DD} = p_D^2$, $p_{Dd} = 2p_D p_d$, $p_{dd} = p_d^2$ and

$$p_D^2 + 2p_D p_d + p_d^2 = 1$$

where p_{DD}, p_{Dd}, p_{dd} are the frequencies for genotypes DD, Dd and dd respectively. If each genotype has a distinct phenotype, or if one of the alleles, i.e d, is recessive to the other, p_d can be found from the square root of the frequency of the phenotype associated with genotype dd. As D and d are the only two alleles, $p_D = 1 - p_d$.

However there are three alleles for blood type. As there are only 4 distinct phenotypes for the 6 possible pairs of alleles (i.e someone with blood type A could have either genotype AA or AO) allele frequencies can not be directly derived from phenotype counts. Thus the need for a method that finds maximum likelihood estimates for the allele frequencies, such as Expectation-Maximization (EM).

EM involves two steps: Find the values of latent variables (or maximum likelihood estimates of them) that can be inferred from observed data and estimates of unknown parameters. Then use the latent variable estimates to make new estimates for the unknown parameters, using functions different to those used in the first step. The two steps are repeated until the new unknown parameter estimates found in the second step are indistinguishable from the estimates used in the first step.

2 Method

In the context of this assignment, the observed data is the phenotype counts $n_A = 15, n_B = 10, n_{AB} = 3, n_O = 1$ for blood types A, B, AB and O respectively; the latent variables are the genotype counts $n_{AA}, n_{AO}, n_{BB}, n_{BO}$ for genotypes AA, AO, BB and BO respectively; and the unknown parameters are

the frequencies of alleles A, B and O: p_A, p_B, p_O. The Expectation-Maximization algorithm was implemented in R [Appendix .1]. It works as follows:

1. Input arbitrary allele frequencies p_A, p_B, p_O that sum to 1
2. Assuming HW-Equilibrium, estimate genotype counts for n_{AO} and n_{BO} using:
$$n_{AO} = \frac{2n_A p_A p_O}{2p_A p_O + p_A^2}$$
$$n_{BO} = \frac{2n_B p_B p_O}{2p_B p_O + p_B^2}$$
where N = total sample size. [*Appendix .1 lines 28, 30*]
3. Estimate genotype counts for n_{AA} and n_{BB} using $n_{AA} = n_A - n_{AO}$ and $n_{BB} = n_B - n_{BO}$. [*Appendix .1 lines 29, 31*]
4. Calculate new allele frequency estimates as follows:
$$p_{ANew} = \frac{2n_{AA} + n_{AO} + n_{AB}}{2N}$$
$$p_{BNew} = \frac{2n_{BB} + n_{BO} + n_{AB}}{2N}$$
$$p_{ONew} = \frac{2n_O + n_{AO} + n_{BO}}{2N}$$
[*Appendix .1 lines 33-35*]
5. If the new allele frequency estimates approximately equal the old estimates [*Appendix .1 lines 11-15, 39-42*], output p_A, p_B, p_O. Else replace the values for p_A, p_B, p_O with the values for $p_{ANew}, p_{BNew}, p_{ONew}$ respectively and repeat steps 2 to 5. [*Appendix .1 lines 43-45*]

EM algorithms potentially get stuck in local maxima instead of finding the global maximum likelihood estimate, depending on the initial input values. To see if this occurred for this implementation, the algorithm was run 50 times with different random initial allele frequency inputs [*Appendix .1 lines 56-75*]. It is possible that a too broad definition of "approximately equal" may result in EM terminating too early as there may be some iterations before the values with maximum likelihood are found where the differences in parameter estimates are small. To prevent this, estimates had to be within 1^{-15} of each other to be considered approximately equal [*Appendix .1 line 37*].

3 Results

All 50 executions output the same estimates for each allele frequency:

$$p_A = 0.4144,\ p_B = 0.2777,\ p_O = 0.3079$$

The median count of iterations through the EM loop until allele frequency estimates converged was 43 and the mean 43.2.

4 Conclusion

The fact that all inputs tried output the same frequency estimates indicates that the functions used to find the estimate do not have maxima other than the global maximum. If this is the case then Expectation-Maximisation is an effective approach to finding allele frequencies from a sample of phenotype counts, not only for blood types but for other traits where phenotypes are determined by a single genetic locus with multiple observed alleles.

Of note is the comparison between phenotype frequencies calculated from these maximum likelihood estimates using HW and the observed phenotype frequencies.

Phenotype	Estimated frequency		Observed frequency	
	Calculation	Value	Calculation	Value
A	$p_A^2 + 2p_Ap_O$	0.4269	$n_A \div N$	0.5172
B	$p_B^2 + 2p_Bp_O$	0.2482	$n_B \div N$	0.3448
AB	$2p_Ap_B$	0.2302	$n_{AB} \div N$	0.1034
O	p_O^2	0.0948	$n_O \div N$	0.0345

As can be observed in the above table the frequencies do not match, with AB being at much lower frequency than expected. This is possibly due to the sample being out of Hardy-Weinberg equilibrium. Further study would be needed to find which of the assumptions behind HW are not holding, and the EM model would need to be adjusted accordingly. However, this could just be sample variance and another larger sample may have phenotype frequencies closer to those estimated by the model.

5 Appendix

.1 R Scrpt

```
nA <- 15
nB <- 10
nAB <- 3
nO <- 1
N <- nA +nB + nAB + nO
realPhenFreqA <- nA / N
realPhenFreqB <- nB / N
realPhenFreqAB <- nAB / N
realPhenFreqO <- nO / N

# Function used by EM to test if new parameter estimate is acceptibly
# close to previous parameter estimate
closeEnough <- function(n1, n2, closeness) {
```

```
        return(n1 > n2 - closeness & n1 < n2 + closeness)
}

# Expectation Maximization for blood type allele frequencies
# Takes a vector of initial allele frequency guesses (pInit) and observed
# phenotype counts (nA, nB, nAB, nO)
emBlood <- function(pInit, nA, nB, nAB, nO) {
        if(sum(pInit) != 1) stop("Initial allele frequencies do not sum to 1")
        pA <- pInit[1]
        pB <- pInit[2]
        pO <- pInit[3]
        N <- nA +nB + nAB + nO
        count <- 1
        while(TRUE){
                nAO <- (2* pA * pO * nA) / (pA**2 + 2* pA * pO)
                nAA <- nA - nAO
                nBO <- (2* pB * pO * nB) / (pB**2 + 2* pB * pO)
                nBB <- nB - nBO

                newPA <- (2*nAA + nAO + nAB) / (2*N)
                newPB <- (2*nBB + nBO + nAB) / (2*N)
                newPO <- (2*nO + nBO + nAO) / (2*N)

                closeness <- 1e-15

                if (!closeEnough(newPA, pA, closeness)
                        || !closeEnough(newPB, pB, closeness)
                        || !closeEnough(newPO, pO, closeness))
                {
                        pA <- newPA
                        pB <- newPB
                        pO <- newPO
                        count <- count + 1
                } else {
                        break
                }
        }
        alleleFreq <- c(pA, pB, pO, count)
        names(alleleFreq) <- c("pA", "pB", "pO", "Iterations until Convergence")
        return(alleleFreq)
}

# Create 50 random sets of initial allele frequencies
# Use to test if different inputs result in finding different local maxima
startfreq <- matrix(nrow = 50, ncol = 3)
for(i in 1:50){
```

```
        ps <- c(0,0,0)
        place <- sample(1:3, 3)
        bar <- runif(2, 0, 1)
        ps[place[1]] <- min(bar)
        ps[place[2]] <- max(bar) - min(bar)
        ps[place[3]] <- 1-max(bar)
        startfreq[i,] <- ps
}
# Check distribution of starting frequencies
# hist(startfreq[,2])

# Run EM on all input sets. Tally each output value for each frequency
maxLAlleleFreq <- apply(startfreq, 1, emBlood,
                        nA = nA, nB = nB, nO = nO, nAB = nAB)
maxLAlleleFreq <- t(maxLAlleleFreq)
freqCounts <- list(table(maxLAlleleFreq[,1]), table(maxLAlleleFreq[,2]),
table(maxLAlleleFreq[,3]))
names(freqCounts) <- colnames(maxLAlleleFreq[,1:3])
# Check median and mean times through loop untill convergence
medianRuns <- median(maxLAlleleFreq[,4])
meanRuns <- mean(maxLAlleleFreq[,4])

# See how phenotype frequencies calculated from an EM allele frequency
# estimate compares to the observed phenotype frequencies
pA <- maxLAlleleFreq[1,1]
pB <- maxLAlleleFreq[1,2]
pO <- maxLAlleleFreq[1,3]
fAEst <- 2* pA * pO + pA**2
fBEst <- 2* pB * pO + pB**2
fABEst <- 2 * pA * pB
fOEst <- pO**2

estimated <- c(fAEst, fBEst, fABEst, fOEst)
observed <- c(realPhenFreqA, realPhenFreqB, realPhenFreqAB, realPhenFreqO)
comparison <- cbind(estimated, observed)
row.names(comparison) <- c('A', 'B', 'AB', 'O')
```