Estimating ABO-Gene Allele Frequencies in Population using Phenotypic Blood-Type Data.

Faizan Khalid Mohsin

September 22, 2020

# Introduction

The ABO-gene (ABO locus) on chromosome 9 can have 3 alleles (antigens: A, B, O). Different pairings of these alleles (AA, AB, BB, AO, etc.) lead to four phenotypic manifestations in the form of 4 blood types: A, B, AB, O.

We want to estimate the frequency of the 3 alleles A, B, O in the population. One way to do this would be to take blood samples of a big sample of people from the population of interest and using DNA sequencing to determine the allele type (A or B or O) on the ABO-gene on chromosome 9. However, to get reliable frequency estimates one needs to do to DNA sequency for a large number of people, which will be extremely expensive and time consuming. A more practical and more efficient method for estimating the 3 allele frequencies of a population would be to collect the phenotypic data, the blood type A, B, AB or O of people, which would be much more cheaper and easier to do at a large scale and using the theoretical work to estimate the 3 allele frequencies in the population using the phenotypic blood type data.

Futher, also in a lab, it could be that we have the blood type of a sample from a population. In such a case a method for estimating the allele frequency in the population will be very useful.

In this paper we will layout the theoretical framework for estimating the allele frequency using phenotypic blood type data and use two separate algorithms and approaches for the estimations. Finally, we will use the actual blood type data gathered from a sample of 2114 people to demonstrate and estimate the allele frequency using the two algorithms.

# Method

We use two methods for approximating the allele frequencies: Expectation-Maximization method and Newton-Raphson method.

In the ideal case we could sample n people and find out how many people possess allele A (nA), how many possess allele B (nB) and how many possess allele (nO). In such a case it would be very easy to calculate the allele frequency in the population: freq(A) = nA/n, freq(B) = nB/n, and freq(O) = nO/n. And we would be done. However, getting nA, nB, nO directly from DNA sequencing is very expensive and time consuming. It is much easier to collect the blood type of each individual from the sample of n people. This will give us a numeric count of how many people have each blood type, namely, n\_A: number of people with blood type A, n\_B: number of people with blood type B, n\_AB: number of people with blood type AB and n\_O: number of people with blood type O. We use genetics to link the number of individuals with alleles nA, nB, and nO with number of people with phenotypes n\_A, n\_B, n\_AB, and n\_O (blood-type).

## Genetical Theory and Notation:

Going from allele frequency to genotypic frequency and going from genotypic frequency to phenotypic frequency.

From genetic theory we know that alleles A, B are dominant to allele O; alleles A, B are co-dominant; and allele O is recessive to alleles A, B. Using this below is a mapping from genotype to phenotype:

|  |  |
| --- | --- |
| Genotype | Phenotype |
| AA or AO | A |
| BB or BO | B |
| AB | AB |
| OO | 0 |

For estimating the frequency of the three alleles let:

Using Hardy–Weinberg equilibrium we get the following:

Using the allele to genotypic data we get the following formulas for the 4 phenotypic frequency:

ALthough, not all populations satisfy the HWE completely, it is still a good approximtion and a reasonable assumption to make.

## Statistical Theory and Likelihood function:

From the notes of Professor Lei Sun, using the previous notation and the Hardy-Weinberg Equilibrium the log likelihood function is:

We can find the maxima’s of this by taking the first derivative, equating to zero and then solving for p and q. However, this is very difficult to do analytically. Hence, we will use two different algorithms and approaches to estimate p and q. Namely, the Newton-Raphson algorithm and the Estimation-Maximization algorithm.

### Newton-Raphson Algorithm Method

We directly find the maxima’s of the loglikelihood function using numeric computational method called Newton-Raphson algorithm.

To approximate the values of

We need to get the full first derivative of this with respect to p and q.

A numerical iterative approach to obtain the maximum (or the minimum) a function: , , e.g.

It is based on the first derivatives (gradient vector), e.g.

Before taking the derivative, the log-likelihood function can be simplied, making it very easy to take the derivative:

Using the properties of log and simple arithmetic we can further simplify the log-likelihood to the following:

Now using the chain rule and the fact that the derivative of is , it is very easy to get the partial derivates with respect to p and q.

where:

and the second derivatives (Hessian matrix):

where the second derivatives which are the components of the hessian are as follows:

Using again the chain rule and the fact that the derivative of is the following second partial derivatives are found:

#### The Newton-Raphson algorithm:

Choose a starting value, .

For the updating function is

Under certain conditions, converges to the value that maximizes (or minimizes) the function.

A few notes on the Newton-Raphson algorithm.

The staring value, , is important: the algorithm is not guaranteed to converge from all starting values, particularly in regions where the matrix is not positive definite.

(Staring values may be obtained from some crude parameter estimates.)

### Estimation-Maximization Algorithm Method

Another method to solving this is to think of the allele frequencies as latent variables or missing variables. In this approach we can cast the problem in the framework of EM algorithm.

To implement the EM algorithm we reframe the question in terms of a missing data problem.

The Expectation-Maximization (EM) algorithm is a numerical iterative method for finding the Maximum Likelihood Estimates (MLE) of parameters.

EM algorithms are often used in situations where the problem of estimation can be solved much easier if certain additional pieces of data are available.

The ABO-blood problem can be formulated as such incomplete data or missing data problem:

Some of the counts of the 6 genotypes are missing: among blood type A: , among blood type B: .

Complete data: , , , , , .

Observed data: , , , .

Missing data: or , or .

and .

E-step: the expected value of the log likelihood is calculated (when the log likelihood is linear w.r.t. to the missing data as in this case, then essentially, the missing data are imputed), assuming some initial values for the parameters, e.g. given the initial parameter values , :

M-step: MLE can then be calculated based on

e.g. MLE of the parameters of interest, and , given the imputed missing data (, , , ), and the observed data (, ):

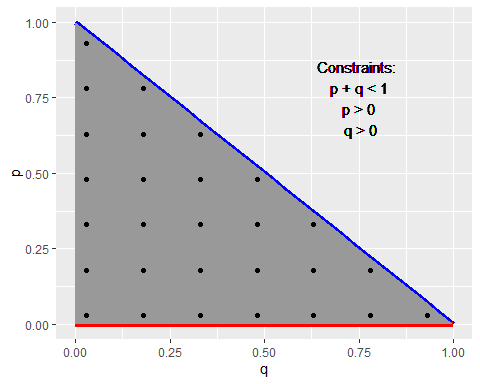
where , the total number of individuals in the sample.

and are improved estimates of the parameters!

Use and to perform the E-step again, and then perform the M-step to obtain improved estimates, and .

Continue until convergence: the changes in parameter estimates (, ) are negligible for the purpose of the study.

### Sample Space of parameters and initial values.



### Selecting Threshold and Accuraccy of Estimates.

We select different threshold values for the accuracy of the estimates. For any praticle purpose one would realistically not need an accuracy of greater than six decimale points . Further, the minimum accuracy one would need would be at least of two decimale points . Keeping this range in mind we implement the two algorithms for threshold values of . These values will allow us to see how the two algo

how the algorithms’ scale with increasing accuracy requirement.

## Data

Table 1: Blood-Type and their counts in the sample population.

|  |  |  |
| --- | --- | --- |
| Blood-Type | Count | Frequency |
| A | 9123 | 0.43 |
| B | 2987 | 0.14 |
| AB | 1269 | 0.06 |
| O | 7725 | 0.37 |
| Total | 21104 | 1.00 |

# Analysis, Results and Code.

## Newton-Raphson Algorithm

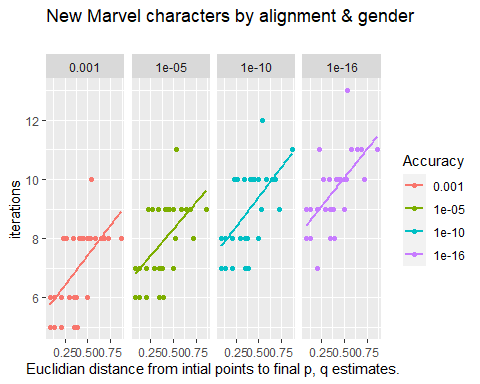
## Time difference of 0.4899969 secs

## [1] 112 10

Table 2: Grouping estimates and number of iterations based on accuracy.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Accuracy | n | mean number of iterations | min iterations | max iterations | mean p estimate | mean q estimate |
| 1e-16 | 27 | 9.629630 | 7 | 13 | 0.2876856 | 0.1065550 |
| 1e-10 | 27 | 8.962963 | 7 | 12 | 0.2876856 | 0.1065550 |
| 1e-05 | 27 | 7.925926 | 6 | 11 | 0.2876856 | 0.1065550 |
| 0.001 | 27 | 7.000000 | 5 | 10 | 0.2876857 | 0.1065543 |

## `geom\_smooth()` using formula 'y ~ x'



## EM Algorithm

## Time difference of 0.279994 secs

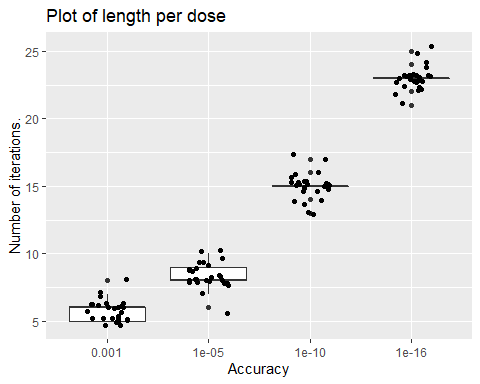
## Factor w/ 4 levels "0.001","1e-05",..: 1 1 1 1 1 1 1 1 1 1 ...

## [1] "factor"

## Factor w/ 4 levels "0.001","1e-05",..: 1 1 1 1 1 1 1 1 1 1 ...

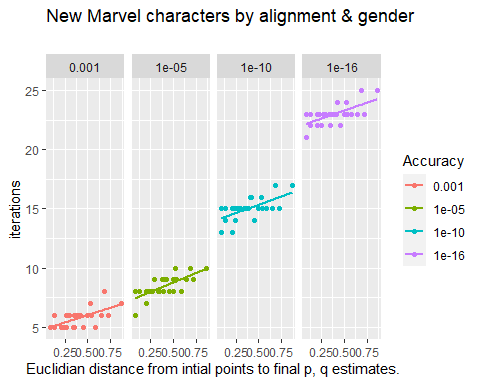
Table 3: Grouping estimates and number of iterations based on accuracy.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Accuracy | n | mean number of iterations | min iterations | max iterations | mean p estimate | mean q estimate |
| 1e-16 | 28 | 22.964286 | 21 | 25 | 0.2876856 | 0.1065550 |
| 1e-10 | 28 | 15.000000 | 13 | 17 | 0.2876856 | 0.1065550 |
| 1e-05 | 28 | 8.392857 | 6 | 10 | 0.2876860 | 0.1065550 |
| 0.001 | 28 | 5.714286 | 5 | 8 | 0.2877468 | 0.1065609 |



We observe that the higher the accuracy (threshold value) the more iterations it takes.

## `geom\_smooth()` using formula 'y ~ x'



Within each accuracy, we notice the further the initial value that we randomly choose is from the final convergence point the higher the number of itereations it takes to converge to the estimates: and .

Furhter, we again notice that the higher the accuracy the more iterations the algorithm takes to converge.

## Comparing Expectation-Maximization Algorithm with Newton-Raphson Algorithm.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| epsilon\_p | n | mean\_num\_iterations | min\_num\_iter | max\_num\_iter | mean\_p | mean\_q |
| 1e-16 | 28 | 22.964286 | 21 | 25 | 0.2876856 | 0.1065550 |
| 1e-10 | 28 | 15.000000 | 13 | 17 | 0.2876856 | 0.1065550 |
| 1e-05 | 28 | 8.392857 | 6 | 10 | 0.2876860 | 0.1065550 |
| 0.001 | 28 | 5.714286 | 5 | 8 | 0.2877468 | 0.1065609 |
| 1e-16 | 27 | 9.629630 | 7 | 13 | 0.2876856 | 0.1065550 |
| 1e-10 | 27 | 8.962963 | 7 | 12 | 0.2876856 | 0.1065550 |
| 1e-05 | 27 | 7.925926 | 6 | 11 | 0.2876856 | 0.1065550 |
| 0.001 | 27 | 7.000000 | 5 | 10 | 0.2876857 | 0.1065543 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| epsilon\_p | n | mean\_num\_iterations | min\_num\_iter | max\_num\_iter | mean\_p | mean\_q | epsilon\_p | n | mean\_num\_iterations | min\_num\_iter | max\_num\_iter | mean\_p | mean\_q |
| 1e-16 | 28 | 22.964286 | 21 | 25 | 0.2876856 | 0.1065550 | 1e-16 | 27 | 9.629630 | 7 | 13 | 0.2876856 | 0.1065550 |
| 1e-10 | 28 | 15.000000 | 13 | 17 | 0.2876856 | 0.1065550 | 1e-10 | 27 | 8.962963 | 7 | 12 | 0.2876856 | 0.1065550 |
| 1e-05 | 28 | 8.392857 | 6 | 10 | 0.2876860 | 0.1065550 | 1e-05 | 27 | 7.925926 | 6 | 11 | 0.2876856 | 0.1065550 |
| 0.001 | 28 | 5.714286 | 5 | 8 | 0.2877468 | 0.1065609 | 0.001 | 27 | 7.000000 | 5 | 10 | 0.2876857 | 0.1065543 |

# Discussion

## Comparing Algorithm Speed and Efficiency.

## Comparing Algorithms’ robustness to initial vaules.

The sample space of the paramters is p, q > 0 and p + q < 1. However, we need to be away from the boundry of the sample space since near or at the boundry of the sample space the Hessian Matrix can become sigular. To see the robustness to initial values of the two algorithms we start from different initial values. We choose these values such that one point is taken from each region and the sample space is covered.

## Choosing the threshold for the stopping criteria.

The threshold can also be thought of as the accuracy we are fine with the estimates.

First observation concerning the choice of threshold values is that the smaller we make this, (the higher the accuray we demand), the more iterations the algorithms take to reach as is expected.

In terms of how small the threshold should be and how much accuracy should be demanded, is a trade between number of iterations and computational cost versus accuracy of estimate.

In partical terms, we would never need an accuracy of greater than 5 decimal points when estimating the allele frequency in the population. Similary, the minimum accuracy we would need would be at least 3 decimal points, anything less would not be a very stable estimate. The number of iterations increases by … from an accuracy of 3 decimal points to 5 decimal points.

We can also see that the

## Computational Time.

In terms of computational time tf\_nr was faster by -42.9%

## Choosing stopping criteria.

For the stopping criteria for the implementation of the two algorthims we took the difference in absolute value between sequent estimates. The distance measured in absolute values is called manhatten distance. However, we could also look if the euclidian distance (L1, or L2) is used instead. We know that euclidian distance is less than or equal to manhattan distance.

However, this is not the most robust stopping criteria, because it is possible that the subsequent change in estimates happens to be smaller than the indicated threshold. One way to . Another thing that can improve the robustness of the stopping criteria is to see check if the rate of

Another thing that could improve the robustness is that instead of looking at only two sequential estimates, to look at several sequential estimates and

Another, thind to keep in mind is that, in several algorithms, there is some randomness in built in the process. Hence, a stopping criteria that is based on several sequential points of estimates and calculating two averages and see if the decrease is small enough. This would be more stable, since two sequential point estimates can be very close randomly, but the genenral trend of the estimate points decrease is not small enough for a given threshold or accuracy. Hence, it is a more robust stopping criteria to look at a set of sequential points and comparing if the average decrease between the two sets of sequential points is less than a given threshold.

## Advantages and Disadvantages of the two Algorithms.

### Advantages of Expectation Maximization Algorithm

### Disadvantage of Expectation Maximization Algorithm

### Advantages of Newton Raphson Algorithm

### Disadvantage of Newton Raphson Algorithm.

One disadvantage in both algorithms is that it is possible that the algorithm converges to a local instead of global minimum or maximum.

One way to determine if we have converged to a global maximum is to do a dense search of the sample space and plot the graph of the log-likelihood. This will

## EM algorithm:

Under regular conditions, the algorithm converges to a local mode of the posterior density.

The rate at which the EM algorithm converges depends on the proportion of missing ``information’’.

## NR algorithm:

The advantage of Newton’s method is: once the iterates are close to the solution, convergence is extremely fast.

If the iterations do not converge: they typically move off quickly toward the edge of the parameter space.

(The remedy can be trying again with a new staring point.)

The computational load can be heavy, if the number of parameters is large, because of the inverse of the Hessian matrix.

# References

# Appendix