

THE HARDY-WEINBERG PRINCIPLE AND ESTIMATING ALLELE FREQUENCIES

Introduction

To keep things relatively simple, we'll spend much of our time in the first part of this course talking about variation at a single genetic locus, even though alleles at many different loci are involved in expression of most morphological or physiological traits. We'll also spend most of our time thinking about only two alleles at that one locus.¹ Towards the end of the course, we'll study the genetics of continuous (quantitative) variation, but until then you can assume that I'm talking about variation at a single locus unless I specifically say otherwise.²

The genetic composition of populations

When I talk about the genetic composition of a population, I'm referring to three aspects of genetic variation within that population:³

1. The number of alleles at a locus.
2. The frequency of alleles at the locus.
3. The frequency of genotypes at the locus.

¹I used to apologize for spending so much time thinking about loci that had only two alleles, but I feel less apologetic now. Much of the population genetic data that is now being gathered derives from single-nucleotide polymorphisms, which are typically polymorphisms involving only two alleles.

²You'll see in a week or a week and a half when we talk about analysis of population structure that we start discussing variation at many loci. But you'll also see that in spite of discussing variation at many loci simultaneously, virtually all of the underlying mathematics is based on the properties of those loci considered one at a time.

³At each locus I'm talking about. Remember, I'm only talking about one locus at a time, unless I specifically say otherwise. We'll see why this matters when I outline the ideas behind genome-wide association mapping towards the end of the course.

It may not be immediately obvious why we need both (2) and (3) to describe the genetic composition of a population, so let me illustrate with two hypothetical populations:

	A_1A_1	A_1A_2	A_2A_2
Population 1	50	0	50
Population 2	25	50	25

It's easy to see that the frequency of A_1 is 0.5 in both populations,⁴ but the genotype frequencies are very different. In point of fact, we don't need both genotype and allele frequencies. We could get away with only genotype frequencies, since we can always calculate allele frequencies from genotype frequencies. But there are fewer allele frequencies than genotype frequencies—only one allele frequency when there are two alleles at a locus. So working with allele frequencies is more convenient when we can get away with it. The challenge is that we can't get genotype frequencies from allele frequencies unless . . .

Derivation of the Hardy-Weinberg principle

We saw last time using the data from *Zoarces viviparus* that we can describe empirically and algebraically how genotype frequencies in one generation are related to genotype frequencies in the next. Let's explore that a bit further. To do so we're going to use a technique that is broadly useful in population genetics,⁵ i.e., we're going to construct a mating table. A mating table consists of three components:

1. A list of all possible genotype pairings.
2. The frequency with which each genotype pairing occurs.
3. The genotypes produced by each pairing.

⁴ $p_1 = 2(50)/200 = 0.5$, $p_2 = (2(25) + 50)/200 = 0.5$.

⁵Although to be honest, we won't see mating tables again after the first couple weeks of the semester.

Female × Male	Frequency	Offspring genotype		
		A_1A_1	A_1A_2	A_2A_2
$A_1A_1 \times A_1A_1$	x_{11}^2	1	0	0
$A_1A_2 \times A_1A_1$	$x_{11}x_{12}$	$\frac{1}{2}$	$\frac{1}{2}$	0
$A_2A_2 \times A_1A_1$	$x_{11}x_{22}$	0	1	0
$A_1A_2 \times A_1A_2$	$x_{12}x_{11}$	$\frac{1}{2}$	$\frac{1}{2}$	0
$A_1A_2 \times A_1A_2$	x_{12}^2	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
$A_2A_2 \times A_1A_2$	$x_{12}x_{22}$	0	$\frac{1}{2}$	$\frac{1}{2}$
$A_2A_2 \times A_1A_1$	$x_{22}x_{11}$	0	1	0
$A_2A_2 \times A_1A_2$	$x_{22}x_{12}$	0	$\frac{1}{2}$	$\frac{1}{2}$
$A_2A_2 \times A_2A_2$	x_{22}^2	0	0	1

Notice that I've distinguished matings by both maternal and paternal genotype. While it's not necessary for this example, we will see examples later in the course where it's important to distinguish a mating in which the female is A_1A_1 and the male is A_1A_2 from ones in which the female is A_1A_2 and the male is A_1A_1 . You are also likely to be surprised to learn that just in writing this table we've already made three assumptions about the transmission of genetic variation from one generation to the next:

Assumption #1 Genotype frequencies are the same in males and females, e.g., x_{11} is the frequency of the A_1A_1 genotype in both males and females.⁶

Assumption #2 Genotypes mate at random *with respect to their genotype at this particular locus*.

Assumption #3 Meiosis is fair. More specifically, we assume that there is no segregation distortion; no gamete competition; no differences in the developmental ability of eggs, or the fertilization ability of sperm.⁷ It may come as a surprise to you, but there are alleles at some loci in some organisms that subvert the Mendelian rules, e.g., the t allele in house mice, segregation distorter in *Drosophila melanogaster*, and spore killer in *Neurospora crassa*.⁸

⁶It would be easy enough to relax this assumption, but it makes the algebra more complicated without providing any new insight, so we won't bother with relaxing it unless someone asks. This is just the first of many examples where I present the math, but I keep it as simple as I possibly can.

⁷We are also assuming that we're looking at offspring genotypes at the zygote stage, so that there hasn't been any opportunity for differential survival.

⁸If you're interested, a pair of papers describing work on spore killer in *Neurospora* appeared in 2012 [3, 4].

Now that we have this table we can use it to calculate the frequency of each genotype in newly formed zygotes in the population,⁹ provided that we're willing to make three additional assumptions:

Assumption #4 There is no input of new genetic material, i.e., gametes are produced without mutation, and all offspring are produced from the union of gametes within this population, i.e., no migration from outside the population.

Assumption #5 The population is of infinite size so that the actual frequency of matings is equal to their expected frequency and the actual frequency of offspring from each mating is equal to the Mendelian expectations.

Assumption #6 All matings produce the same number of offspring, on average.

Taking these three assumptions together allows us to conclude that the frequency of a particular genotype in the pool of newly formed zygotes is

$$\sum (\text{frequency of mating})(\text{frequency of genotype produce from mating}) \quad .$$

So

$$\begin{aligned} \text{freq.}(A_1A_1 \text{ in zygotes}) &= x_{11}^2 + \frac{1}{2}x_{11}x_{12} + \frac{1}{2}x_{12}x_{11} + \frac{1}{4}x_{12}^2 \\ &= x_{11}^2 + x_{11}x_{12} + \frac{1}{4}x_{12}^2 \\ &= (x_{11} + x_{12}/2)^2 \\ &= p^2 \\ \text{freq.}(A_1A_2 \text{ in zygotes}) &= 2pq \\ \text{freq.}(A_2A_2 \text{ in zygotes}) &= q^2 \end{aligned}$$

Those frequencies probably look pretty familiar to you. They are, of course, the familiar Hardy-Weinberg proportions. But we're not done yet. In order to say that these proportions will also be the genotype proportions of adults in the progeny generation, we have to make two more assumptions:

Assumption #7 Generations do not overlap.¹⁰

Assumption #8 There are no differences among genotypes in the probability of survival.

⁹Not just the offspring from these matings.

¹⁰Or the allele frequency is the same in generations that do overlap.

The Hardy-Weinberg principle

After a single generation in which *all* eight of the above assumptions are satisfied

$$\text{freq.}(A_1A_1 \text{ in adults}) = p^2 \quad (1)$$

$$\text{freq.}(A_1A_2 \text{ in adults}) = 2pq \quad (2)$$

$$\text{freq.}(A_2A_2 \text{ in adults}) = q^2 \quad (3)$$

It's vital to understand the logic here.

1. If Assumptions #1–#8 are true, then equations 1–3 **must** be true.
2. If genotypes are *not* in Hardy-Weinberg proportions, one or more of Assumptions #1–#8 **must** be false.
3. If genotypes are in Hardy-Weinberg proportions, one or more of Assumptions #1–#8 may still be violated.
4. Assumptions #1–#8 are *sufficient* for Hardy-Weinberg to hold, but they are not *necessary* for Hardy-Weinberg to hold.

Point (2) is why the Hardy-Weinberg principle is so important. There isn't a population of any organism anywhere in the world that satisfies all 8 assumptions, even for a single generation.¹¹ But *all* possible evolutionary processes within populations cause a violation of at least one of these assumptions. Departures from Hardy-Weinberg are one way in which we can detect those processes and estimate their magnitude.¹²

Estimating allele frequencies

Before we can determine whether genotypes in a population are in Hardy-Weinberg proportions, we need to be able to estimate the frequency of both genotypes and alleles. This is easy when you can identify all of the alleles within genotypes, but suppose that we're trying to estimate allele frequencies in the ABO blood group system in humans. Then we have a situation that looks like this:

¹¹There may be some that come reasonably close, but none that fulfill them *exactly*. There aren't any populations of infinite size, for example.

¹²Actually, there's a ninth assumption that I didn't mention. Everything I said here depends on the assumption that the locus we're dealing with is autosomal. We can talk about what happens with sex-linked loci, if you want. But again, mostly what we get is algebraic complications without a lot of new insight.

Phenotype	A	AB	B	O
Genotype(s)	aa ao	ab	bb bo	oo
No. in sample	N_A	N_{AB}	N_B	N_O

Now we can't directly count the number of a , b , and o alleles. What do we do? Well, nearly 70 years ago, some geneticists figured out one approach with a method they called "gene counting" [1] and that statisticians later generalized for a wide variety of purposes and called the EM algorithm [2]. It uses a trick you'll see repeatedly through this course. When we don't know something we want to know, we pretend that we know it and do some calculations with what we just pretended to know. If we're lucky, we can fiddle with our calculations a bit to relate the thing that we pretended to know to something we actually do know so we can figure out what we wanted to know. Make sense? Probably not. Let's try an example and see if that helps.

If we knew p_a , p_b , and p_o , we could figure out how many individuals with the A phenotype we *expect* to have the aa genotype and how many we *expect* to have the ao genotype, namely

$$\begin{aligned} N_{aa} &= n_A \left(\frac{p_a^2}{p_a^2 + 2p_a p_o} \right) \\ N_{ao} &= n_A \left(\frac{2p_a p_o}{p_a^2 + 2p_a p_o} \right) . \end{aligned}$$

Obviously we could do the same thing for the B phenotype:

$$\begin{aligned} N_{bb} &= n_B \left(\frac{p_b^2}{p_b^2 + 2p_b p_o} \right) \\ N_{bo} &= n_B \left(\frac{2p_b p_o}{p_b^2 + 2p_b p_o} \right) . \end{aligned}$$

Notice that $N_{ab} = N_{AB}$ and $N_{oo} = N_O$ (lowercase subscripts refer to genotypes, uppercase to phenotypes). What we've just done is the "E" part of the EM algorithm, E for "expectation." If we knew all this, then we could calculate p_a , p_b , and p_o from

$$\begin{aligned} p_a &= \frac{2N_{aa} + N_{ao} + N_{ab}}{2N} \\ p_b &= \frac{2N_{bb} + N_{bo} + N_{ab}}{2N} \\ p_o &= \frac{2N_{oo} + N_{ao} + N_{bo}}{2N} , \end{aligned}$$

where N is the total sample size. That’s the “M” part of the EM algorithm, M for “maximization.”¹³

Surprisingly enough we can actually estimate the allele frequencies by using this trick. Just take a guess at the allele frequencies. Any guess will do. Then calculate N_{aa} , N_{ao} , N_{bb} , N_{bo} , N_{ab} , and N_{oo} as described in the preceding paragraph.¹⁴ That’s the **E**xpectation part the EM algorithm. Now take the values for N_{aa} , N_{ao} , N_{bb} , N_{bo} , N_{ab} , and N_{oo} that you’ve calculated and use them to calculate new values for the allele frequencies. That’s the **M**aximization part of the EM algorithm. It’s called “maximization” because what you’re doing is calculating maximum-likelihood estimates of the allele frequencies, given the observed (and made up) genotype counts.¹⁵ Chances are your new values for p_a , p_b , and p_o won’t match your initial guesses, but¹⁶ if you take these new values and start the process over and repeat the whole sequence several times, eventually the allele frequencies you get out at the end match those you started with. These are maximum-likelihood estimates of the allele frequencies.¹⁷

Consider the following example:

Phenotype	A	AB	AB	O
No. in sample	25	50	25	15

We’ll start with the guess that $p_a = 0.33$, $p_b = 0.33$, and $p_o = 0.34$. With that assumption we would calculate that $25(0.33^2/(0.33^2 + 2(0.33)(0.34))) = 8.168$ of the A phenotypes in the sample have genotype aa , and the remaining 16.832 have genotype ao . Similarly, we can calculate that 8.168 of the B phenotypes in the population sample have genotype bb , and the remaining 16.832 have genotype bo . Now that we have a guess about how many individuals of each genotype we have,¹⁸ we can calculate a new guess for the allele frequencies, namely $p_a = 0.362$, $p_b = 0.362$, and $p_o = 0.277$. By the time we’ve repeated this process four more times, the allele frequencies aren’t changing anymore, and the maximum likelihood estimate of the allele frequencies is $p_a = 0.372$, $p_b = 0.372$, and $p_o = 0.256$.

¹³“Maximization of what?” you may ask. “Maximization of the likelihood is the answer, which probably isn’t helpful now, but should be soon.

¹⁴Chances are N_{aa} , N_{ao} , N_{bb} , and N_{bo} won’t be integers. That’s OK. Pretend that there really are fractional animals or plants in your sample and proceed.

¹⁵If you don’t know what maximum-likelihood estimates are, don’t worry. We’ll get to that in a moment.

¹⁶Yes, truth *is* sometimes stranger than fiction.

¹⁷I should point out that this method *assumes* that genotypes are found in Hardy-Weinberg proportions.

¹⁸Since we’re making these genotype counts up, we can also pretend that it makes sense to have fractional numbers of genotypes.

What is a maximum-likelihood estimate?

I just told you that the method I described produces “maximum-likelihood estimates” for the allele frequencies, but I haven’t told you what a maximum-likelihood estimate *is*. The good news is that you’ve been using maximum-likelihood estimates for as long as you’ve been estimating anything, without even knowing it. Although it will take me a while to explain it, the idea is actually pretty simple.

Suppose we had a sock drawer with two colors of socks, red and green. And suppose we were interested in estimating the proportion of red socks in the drawer. One way of approaching the problem would be to mix the socks well, close our eyes, take one sock from the drawer, record its color and replace it. Suppose we do this N times. We know that the number of red socks we’ll get might be different the next time, so the number of red socks we actually get is a random variable. Let’s call that random variable K . Now suppose in our actual experiment we find k red socks, i.e., the value our random variable takes on is k or putting it in an equation: $K = k$. If we knew p , the proportion of red socks in the drawer, we could calculate the probability of getting the data we observed, namely

$$P(K = k|p, N) = \binom{N}{k} p^k (1 - p)^{(N-k)} \quad . \quad (4)$$

This is the *binomial probability distribution*. The part on the left side of the equation is read as “The probability that we get k red socks in our sample *given* the value of p and the sample size N .” The word “given” means that we’re calculating the probability of our data conditional on the (unknown) value p and the (known) sample size N .

Of course we don’t know p , so what good does writing (4) do? Well, suppose we reverse the question to which equation (4) is an answer and call the expression in (4) the “likelihood of the data.” Suppose further that we find the value of p that makes the likelihood bigger than any other value we could pick.¹⁹ Then \hat{p} is the maximum-likelihood estimate of p .²⁰

In the case of the ABO blood group that we just talked about, the likelihood is a bit more complicated

$$\binom{N}{N_A N_{AB} N_B N_O} (p_a^2 + 2p_a p_o)^{N_A} 2p_a p_b^{N_{AB}} (p_b^2 + 2p_b p_o)^{N_B} (p_o^2)^{N_O} \quad (5)$$

This is a *multinomial probability distribution*. It turns out that one way to find the values of p_a , p_b , and p_o is to use the EM algorithm I just described.²¹ There isn’t a simple formula

¹⁹Technically, we treat $P(K = k|p, N)$ as a function of p , find the value of p that maximizes it, and call that value \hat{p} .

²⁰You’ll be relieved to know that in this case, $\hat{p} = k/N$.

²¹There’s another way I’d be happy to describe if you’re interested, but it’s a lot more complicated.

that allows us to write down an expression for the maximum-likelihood estimate of the allele frequencies in terms of the phenotype frequencies. We have to use an algorithm to find them, and the EM algorithm happens to be a particularly convenient algorithm to use.

An introduction to Bayesian inference

Maximum-likelihood estimates have a lot of nice features, but they are also a slightly backwards way of looking at the world. The likelihood of the data is the probability of the data, x , given parameters that we don't know, ϕ , i.e. $P(x|\phi)$. It seems a lot more natural to think about the probability that the unknown parameter takes on some value, given the data, i.e., $P(\phi|x)$. Surprisingly, these two quantities are closely related. Bayes' Theorem tells us that

$$P(\phi|x) = \frac{P(x|\phi)P(\phi)}{P(x)} \quad . \quad (6)$$

We refer to $P(\phi|x)$ as the *posterior distribution* of ϕ , i.e., the probability that ϕ takes on a particular value given the data we've observed, and to $P(\phi)$ as the *prior distribution* of ϕ , i.e., the probability that ϕ takes on a particular value *before* we've looked at any data. Notice how the relationship in (6) mimics the logic we use to learn about the world in everyday life. We start with some prior beliefs, $P(\phi)$, and modify them on the basis of data or experience, $P(x|\phi)$, to reach a conclusion, $P(\phi|x)$. That's the underlying logic of Bayesian inference.

Estimating allele frequencies with two alleles

Let's suppose we've collected data from a population of *Protea repens*²² and have found 7 alleles coding for the *fast* allele at a enzyme locus encoding glucose-phosphate isomerase in a sample of 20 alleles. We want to estimate the frequency of the *fast* allele. The maximum-likelihood estimate is $7/20 = 0.35$, which we got by finding the value of p that maximizes

$$P(k|N, p) = \binom{N}{k} p^k (1-p)^{N-k} \quad ,$$

where $N = 20$ and $k = 7$. A Bayesian uses the same likelihood, but has to specify a prior distribution for p . If we didn't know anything about the allele frequency at this locus in *P. repens* before starting the study, it makes sense to express that ignorance by choosing $P(p)$

²²A few of you may recognize that I didn't choose that species entirely at random, even though the "data" I'm presenting here are entirely fanciful.

to be a uniform random variable on the interval $[0, 1]$. That means we regarded all values of p as equally likely prior to collecting the data.²³

Until the early 1990s²⁴ it was necessary to do a bunch of complicated calculus to combine the prior with the likelihood to get a posterior. Since the early 1990s statisticians have used a simulation approach, Monte Carlo Markov Chain sampling, to construct numerical samples from the posterior. For the problems encountered in this course, we'll mostly be using the freely available software package **Stan** through its interface in **R**, **rstan**, to implement Bayesian analyses. For the problem we just encountered, here's the code that's needed to get our results:²⁵

```
data {
  int<lower=0> N;      // the sample size
  int<lower=0> k;      // the number of A_1 alleles observed
}

parameters {
  real<lower=0, upper=1> p; // the allele frequency
}

model {
  // likelihood
  //
  k ~ binomial(N, p);

  // prior
  p ~ uniform(0.0, 1.0);
}
```

We can run this in **R** by `source()`'ing the following code. Remember that in our fictitious example, we found 7 fast alleles in a sample of 20, i.e., $k = 7$ and $N = 20$.

```
## Load the rstan library
```

²³If we had prior information about the likely values of p , we'd pick a different prior distribution to reflect our prior information. See the Summer Institute notes for more information, if you're interested.

²⁴You are probably thinking to yourself "The 1990s? That's ancient history. Why is Holsinger making such a big deal about this?" Please cut me a little slack. I know that most of you weren't born in the early 90s, but I'd already taught this course two or three times by the time the paper I'm about to refer to was published.

²⁵This code and other **Stan** code used in the course can be found on the course web site by following the links associated with the corresponding lecture.

```

##
library(rstan)

## set the number of chains to the number of cores in the computer
##
options(mc.cores = parallel::detectCores())

## set up the data
## N: sample size
## k: number of A1 alleles
stan_data <- list(N = 20,
                  k = 7)

## Invoke stan
##
fit <- stan("binomial-model.stan",
            data = stan_data,
            refresh = 0)

## print the results on the console with 3 digits after the decimal
##
print(fit, digits = 3)

```

Here's what you'll see in the terminal.²⁶

```

> source("binomial-model.R")
Loading required package: StanHeaders
Loading required package: ggplot2
rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)
For execution on a local, multicore CPU with excess RAM we recommend calling
options(mc.cores = parallel::detectCores()).
To avoid recompilation of unchanged Stan programs, we recommend calling
rstan_options(auto_write = TRUE)
Inference for Stan model: binomial-model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

```

²⁶Your computer may appear to freeze after the message about avoiding recompilation. Don't worry. It's just thinking.

```

      mean se_mean   sd   2.5%   25%   50%   75%   97.5% n_eff  Rhat
p      0.360  0.003 0.099  0.179  0.289  0.357  0.424  0.561 1475 1.001
lp__ -14.926  0.017 0.719 -16.901 -15.088 -14.646 -14.470 -14.421 1691 1.000

```

Samples were drawn using NUTS(diag_e) at Sat Jun 5 16:54:55 2021.
 For each parameter, n_eff is a crude measure of effective sample size,
 and Rhat is the potential scale reduction factor on split chains (at
 convergence, Rhat=1).
 >

Most of the column headings should be fairly self-explanatory. mean is our best guess for the value for the frequency of the *fast* allele, the posterior mean of p . sd is the posterior standard deviation of p . It's our best guess of the uncertainty associated with our estimate of the frequency of the *fast* allele. The 2.5%, 50%, and 97.5% columns are the percentiles of the posterior distribution. The [2.5%, 97.5%] interval is the 95% credible interval, which is analogous to the 95% confidence interval in classical statistics, except that we can say that there's a 95% chance that the frequency of the *fast* allele lies within this interval.²⁷ Since the results are from a simulation, different runs will produce slightly different results. In this case, we have a posterior mean of about 0.36 (as opposed to the maximum-likelihood estimate of 0.35), and there is a 95% chance that p lies in the interval [0.18, 0.56].

Returning to the ABO example

Here's data from the ABO blood group:²⁸

Phenotype	A	AB	B	O	Total
Observed	862	131	365	702	2060

To estimate the underlying allele frequencies, p_A , p_B , and p_O , we have to remember how the allele frequencies map to phenotype frequencies:

$$\begin{aligned}
 \text{Freq}(A) &= p_A^2 + 2p_Ap_O \\
 \text{Freq}(AB) &= 2p_Ap_B \\
 \text{Freq}(B) &= p_B^2 + 2p_Bp_O \\
 \text{Freq}(O) &= p_O^2 \quad .
 \end{aligned}$$

²⁷If you don't understand why that's different from a standard confidence interval, ask me about it.

²⁸This is almost the last time! I promise.

Hers's the Stan code we use to estimate the allele frequencies:

```
data {
  int<lower=0> N_A;
  int<lower=0> N_AB;
  int<lower=0> N_B;
  int<lower=0> N_O;
}

transformed data {
  int<lower=0> N[4];

  N[1] = N_A;
  N[2] = N_AB;
  N[3] = N_B;
  N[4] = N_O;
}

parameters {
  // the three allele frequencies add to 1
  //
  simplex[3] p;
}

transformed parameters {
  real<lower=0, upper=1> p_a;
  real<lower=0, upper=1> p_b;
  real<lower=0, upper=1> p_o;
  // the four phenotype frequencies add to 1
  //
  simplex[4] x;

  // allele frequencies
  //
  p_a = p[1];
  p_b = p[2];
  p_o = p[3];
  // phenotype frequencies
```

```

//
// A
x[1] = p_a^2 + 2*p_a*p_o;
// AB
x[2] = 2*p_a*p_b;
// B
x[3] = p_b^2 + 2*p_b*p_o;
// O
x[4] = p_o^2;
}

model {
  // likelihood
  //
  N ~ multinomial(x);

  // prior
  //
  p ~ dirichlet(rep_vector(1.0, 3));
}

```

The `dirichlet()` prior produces a uniform distribution across all three allele frequencies while ensuring that they sum to 1. Here are the results of the analysis:

```

> source("abo-model.R")
Inference for Stan model: abo-model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
p[1]	0.281	0.000	0.008	0.266	0.276	0.281	0.287	0.297	3814	1.000
p[2]	0.129	0.000	0.005	0.119	0.126	0.129	0.133	0.140	3685	1.000
p[3]	0.589	0.000	0.008	0.573	0.584	0.589	0.595	0.605	3428	1.001
p_a	0.281	0.000	0.008	0.266	0.276	0.281	0.287	0.297	3814	1.000
p_b	0.129	0.000	0.005	0.119	0.126	0.129	0.133	0.140	3685	1.000
p_o	0.589	0.000	0.008	0.573	0.584	0.589	0.595	0.605	3428	1.001
x[1]	0.411	0.000	0.010	0.391	0.404	0.411	0.418	0.431	4033	1.000
x[2]	0.073	0.000	0.003	0.067	0.071	0.073	0.075	0.079	3417	1.001
x[3]	0.169	0.000	0.007	0.156	0.164	0.169	0.174	0.183	3764	1.000
x[4]	0.347	0.000	0.010	0.328	0.341	0.347	0.354	0.366	3430	1.001
lp__	-2506.009	0.022	0.963	-2508.531	-2506.366	-2505.715	-2505.316	-2505.068	1915	1.000

Samples were drawn using NUTS(diag_e) at Sat Jun 5 17:23:22 2021.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
>

The posterior means for the allele frequencies are indistinguishable from the maximum-likelihood estimates ($p_a = 0.281$, $p_b = 0.129$, and $p_o = 0.59$), but we also have 95% credible intervals so that we have an assessment of how reliable the Bayesian estimates are. We also have estimates of the phenotype frequencies and their reliability. Getting estimates of the reliability for the allele frequencies from a likelihood analysis is possible, but it takes a fair amount of additional work.

References

- [1] R Ceppellini, M Siniscalco, and C A B Smith. The estimation of gene frequencies in a random-mating population. *Annals of Human Genetics*, 20:97–115, 1955.
- [2] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.
- [3] Thomas M Hammond, David G Rehard, Hua Xiao, and Patrick K T Shiu. Molecular dissection of *Neurospora* Spore killer meiotic drive elements. *Proceedings of the National Academy of Sciences of the United States of America*, 109(30):12093–12098, July 2012.
- [4] Sven J Saupe. A fungal gene reinforces Mendel’s laws by counteracting genetic cheating. *Proceedings of the National Academy of Sciences of the United States of America*, 109(30):11900–11901, July 2012.

Creative Commons License

These notes are licensed under the Creative Commons Attribution License. To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.