The Metropolis-Hastings algorithm in more than one dimension

Goals of this lecture:

- Introduce a simple model for “inbreeding”
- Consider MCMC in more than a single dimension
- Describe component-wise Metropolis-Hastings sampling

Genotype Frequencies and Inbreeding:

Starting with the last exercise of of the last session: one locus with two alleles, A and a, at frequencies p and $1 - p$, respectively, and "inbreeding coefficient" f.

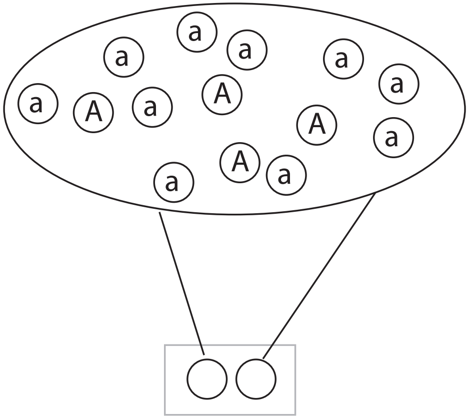
Probabilities of the three genotypes are:

- $P(AA) = fp + (1 - f)p^2$
- $P(Aa \text{ or } aA) = (1 - f)2p(1 - p)$
- $P(aa) = f(1 - p) + (1 - f)(1 - p)^2$

Since "inbreeding" is used to describe a lot of (related) things, let's briefly review what this model is saying. . .

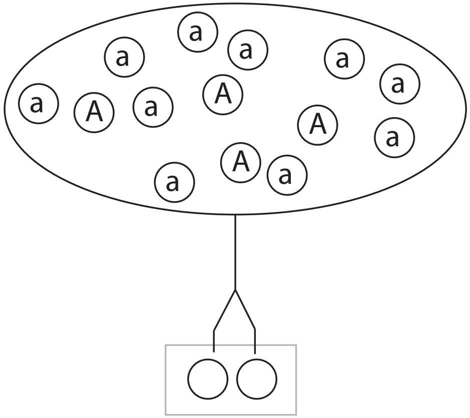
Inbreeding model:

Not-inbred with probability $1-f$



- $P(AA) = p^2$
- $P(Aa \text{ or } aA) = 2p(1-p)$
- $P(aa) = (1-p)^2$

Inbred with probability f



- $P(AA) = p$
- $P(Aa \text{ or } aA) = 0$
- $P(aa) = (1-p)$

$$
\begin{aligned}
P(n_{AA}, n_{Aa}, n_{aa} | p, f) &= C \times [fp + (1-f)p^2]^{n_{AA}} \times [(1-f)2p(1-p)]^{n_{Aa}} \\
&\times [f(1-p) + (1-f)(1-p)^2]^{n_{aa}}
\end{aligned}
$$

What Our Data Would Look Like:

- We have a sample of n individuals total.
  - n_{AA} are homozygous for the A allele
  - n_{Aa} are heterorozygous
  - n_{aa} are homozygous for the a allele
- Clearly, $n = n_{AA} + n_{Aa} + n_{aa}$

A concrete example we will use throughout this lecture is $n = 50$ with:

$$n_{AA} = 30 \qquad n_{Aa} = 10 \qquad n_{aa} = 10$$

which are roughly the expected values if $p = .7$ and $f = .5$.

Bayesian Model for Estimating f and p:

To go about estimating f and p in a Bayesian fashion, we must fully specify the model. This means that we need

- Priors for f and p. We will choose uniform priors $f \sim \mathtt{Beta}(1,1)$ and $p \sim \mathtt{Beta}(1,1)$
- The data themselves. These are n_{AA}, n_{Aa}, and n_{aa}.
- The likelihood: $P(n_{AA}, n_{Aa}, n_{aa}|p, f)$, which is given on the previous page

The posterior distribution is then prior $\times$ likelihood, divided by the ("nasty") normalizing constant

$$P(p, f|n_{AA}, n_{Aa}, n_{aa}) = \frac{P(f)P(p)P(n_{AA}, n_{Aa}, n_{aa}|p, f)}{\int_{f,p} P(f)P(p)P(n_{AA}, n_{Aa}, n_{aa}|p, f)df\,dp}$$

Computing the normalizing constant is difficult, but with MCMC, *we don't have to!*

Recall, to perform MCMC it suffices to know the target distribution up to a constant of proportionality.

Our target distribution is

$$P(p, f|n_{AA}, n_{Aa}, n_{aa}) \propto P(f)P(p)P(n_{AA}, n_{Aa}, n_{aa}|p, f)$$

So long as $0 < f < 1$ and $0 < p < 1$.

Otherwise it is 0.

Two-Dimensional M-H Sampler:

We can compute the target distribution (up to a constant) easily for any f and p. So, to simulate from this posterior distribution we can just implement a Metropolis-Hastings sampler.

Following the examples of the last session, for p we will choose a normal proposal distribution centered on the current value with standard deviation of s_p:

$$q(p^*|p) \equiv \text{Normal}(p, s_p)$$

And, we will use the same for f:

$$q(f^*|f) \equiv \text{Normal}(f, s_f)$$

Applying these proposal distributions in sequence gives us a simple way to simulate proposed values, (p^*, f^*), from the current values (p, f).

A "sweep" of our MCMC algorithm would look like:

1. Propose a new value, (p^*, f^*) for (p, f)
   - propose p^* from $\mathsf{Normal}(p, s_p)$
   - propose f^* from $\mathsf{Normal}(f, s_f)$
2. Accept or reject the proposed value (p^*, f^*) with probability R:

$$R = \min\left\{1, \frac{q(p|p^*)q(f|f^*)}{q(p^*|p)q(f^*|f)} \times \frac{P(p^*)P(f^*)P(n_{AA}, n_{Aa}, n_{aa}|p^*, f^*)}{P(p)P(f)P(n_{AA}, n_{Aa}, n_{aa}|p, f)}\right\}$$

If you accept the proposed value, set the current value to the proposed value. Otherwise leave the current values unchanged.

Computer Demo: `inbred_p -n 30 10 10` (starts on "Jointly" (j))

Component-wise Metropolis Hastings Sampler :

- In any MCMC implementation, the proposal distribution *need not propose changes to* **every** *variable/parameter in the model*.
- In fact, there are few "real-world" problems requiring MCMC in which you would use a single proposal distribution in which changes were proposed to all the variables in the model.
- **Important Concept:**
  - Any proposal distribution, regardless of how many or how few variables it proposes changes to, is valid, so long as the proposal is accepted or rejected in a way that satisfies detailed balance w.r.t. the target distribution (i.e., is done via the Metropolis-Hastings algorithm).
  - These different flavors of the "propose-reject/accept" step may be combined in series in whatever manner is desired, so long as they produce an irreducible, aperiodic chain[1].

[1] Nonetheless, some ways are better than others and will lead to a better-mixing chain

Simple Component-wise M-H Sampler for p and f:

Simplest scenario has a sweep as follows:

1. Do an update for p:
   (a) propose p^* from $\text{Normal}(p, s_p)$
   (b) Accept or reject the proposed value p^* with probability $\min\{1, \alpha\}$, where:

$$\alpha = \frac{q(p|p^*)}{q(p^*|p)} \times \frac{P(p^*)P(f)P(n_{AA}, n_{Aa}, n_{aa}|p^*, f)}{P(p)P(f)P(n_{AA}, n_{Aa}, n_{aa}|p, f)}$$

2. Do an update for f:
   (a) propose f^* from $\text{Normal}(f, s_f)$
   (b) Accept or reject the proposed value f^* with probability $\min\{1, \alpha\}$, where:

$$\alpha = \frac{q(f|f^*)}{q(f^*|f)} \times \frac{P(p)P(f^*)P(n_{AA}, n_{Aa}, n_{aa}|p, f^*)}{P(p)P(f)P(n_{AA}, n_{Aa}, n_{aa}|p, f)}$$

Simple Component-wise M-H Sampler for p and f, cont'd:

Some very important points:

- The proposals are each a little simpler (though just slightly...) than jointly proposing changes to (p, f)
- Neither step 1 nor step 2 of the sweep creates an irreducible chain (obviously, if you never update p, for example, your chain could never reach every possible value of p).
- However, taken together, steps 1 and 2 create an irreducible chain.

Computer Demo: `inbred_p` (Component-wise using (c) from info window)

Simple Component-wise M-H Sampler for p and f, cont'd:

Two more important points

- The form of the Hastings ratio is a little simpler when we have proposed changing just a subset of the variables.
- However, in this case the target density remains just as complex, because it does not factorize into a separate part for f and a part for p.
- Since we are changing just a small part of the model at a time, it seems like we could spend some more energy on making each separate proposal distribution more "intelligent."

The final two points above get us to thinking about Gibbs sampling, which we will return to after a brief discussion of latent variables. . .

Wrap-Up:

Main Points:

- In problems where it is useful, MCMC almost always proceeds by proposing changes to a small subset of the variables.
- There may be many different proposal types.
  - Each proposal type *must* satisfy detailed balance.
  - Each proposal type need not make an irreducible chain, BUT
  - All proposals taken together should form an irreducible chain.