Bayesian statistics for geneticists

Matthieu Vignes
m.vignes@massey.ac.nz

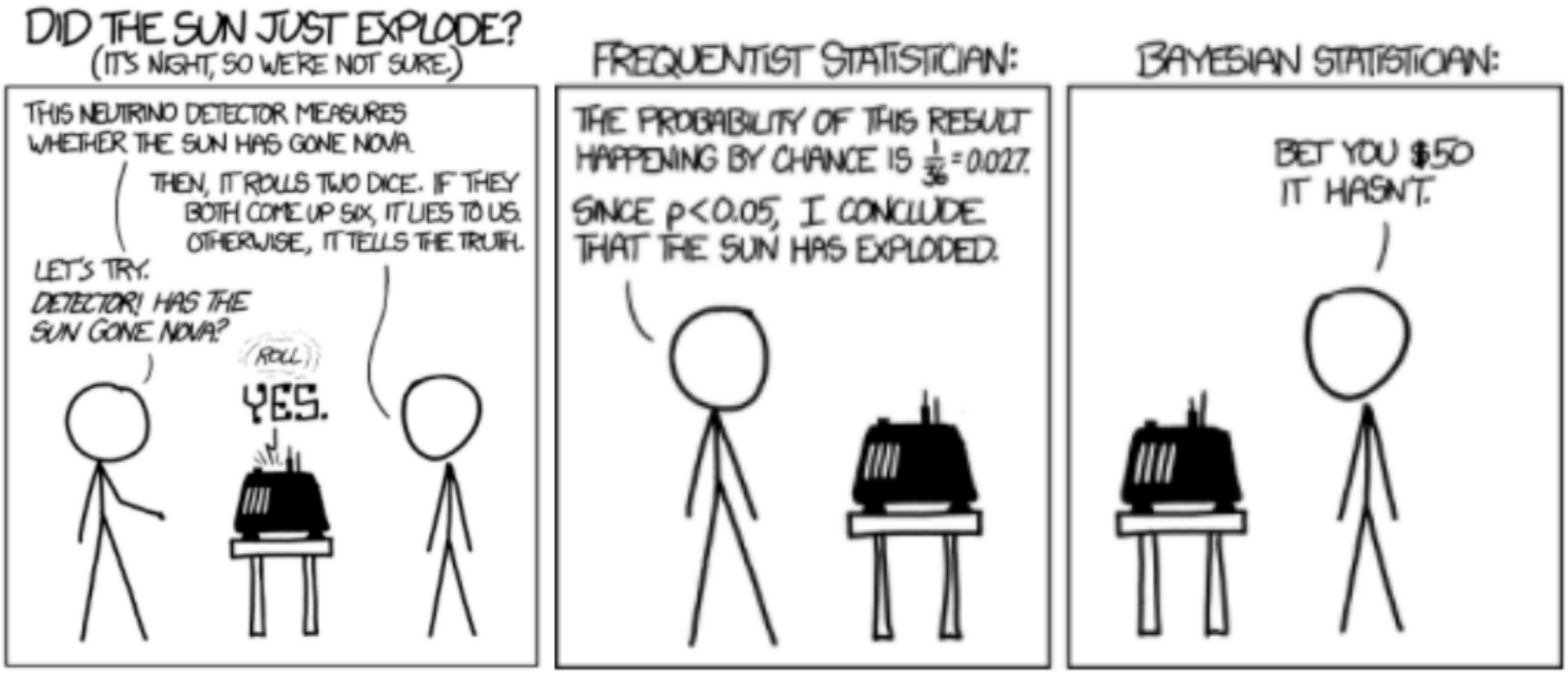
School of Mathematical and Computational Sciences
Massey University

Tuesday 9 to Friday 12 April 2024

Welcome

- This course is an introduction to Bayesian Statistics for geneticists (and bioinformaticians, breeders...).
- Lots of material taken from K. Rice, M. Stephens and E. Anderson. Thx to them!
- You will find additional material here `https://github.com/MatthieuVignes/Lincoln2024`
- I assume basic knowledge of R, but please don't let this stop you. To keep the focus on statistics, the genetic examples are easy.

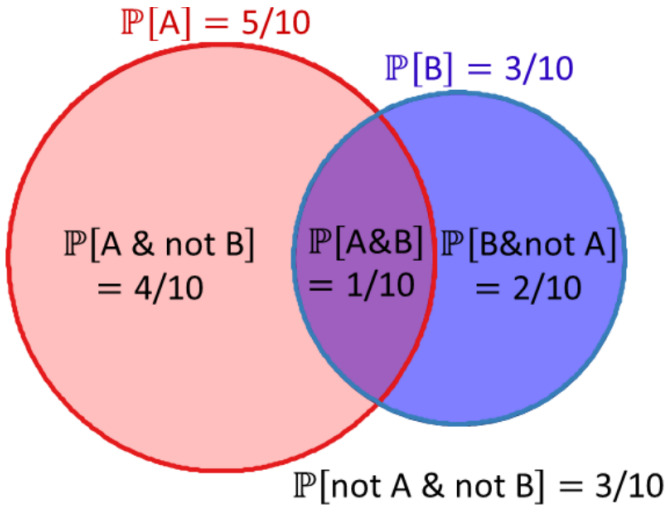
To be Bayesian or not to be



Bayes' Theorem

Graphically

It describes conditional probabilities for events A and B. $P(A|B)$ denotes the probability that A happens **given** that B happens. For example:



$$P(A|B) = \frac{1/10}{3/10} = 1/3 \text{ and}$$

$$P(B|A) = \frac{1/10}{5/10} = 1/5$$

Bayes' Theorem

Equation and numbers

Bayes' theorem states that $P(A|B)$ and $PB|A)$ are related via:

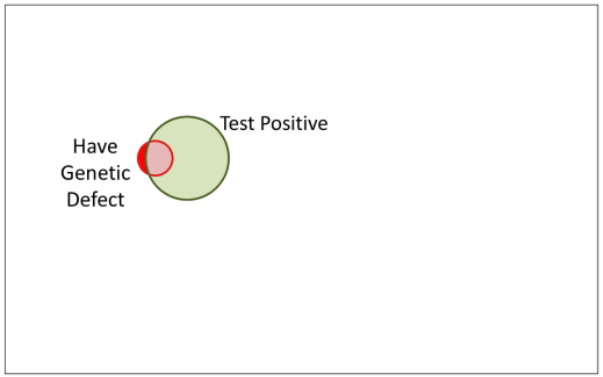
$$P(A|B) = \frac{P(A \cap B}{P(B) = P(B|A)\frac{P(A)}{P(B}}$$

So here $1/3 = 1/5 \times \frac{5/10}{3/10}$, works :)
In words: the conditional probability of A given B is the conditional probability of B given A scaled by the relative probability of A compared to B.

Bayes' Theorem

Screening for a genetic variant

Why does it matter? Well let's see what your intuition tells you. If 1% of the population have a genetic variant you want to detect, with a screening test that has 80% sensitivity and 95% specificity.



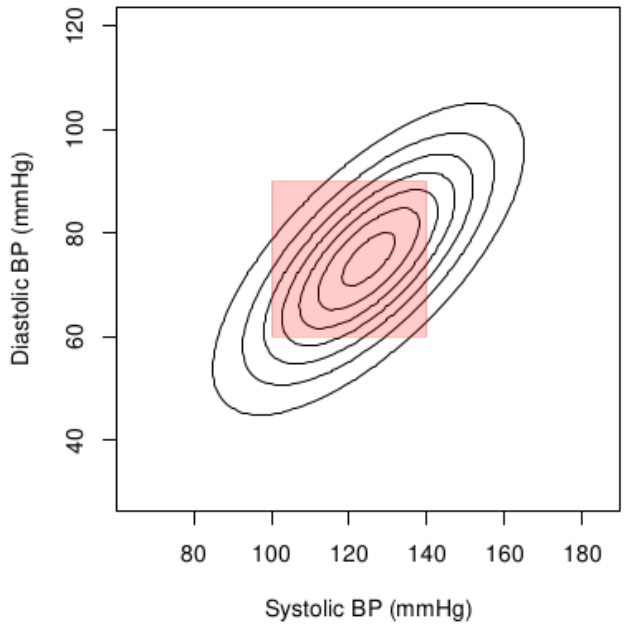
$P(\text{Test}-|\text{no variant}) = 95\%$, $P(\text{Test}+|\text{variant}) = 80\%$ and $\frac{P(\text{Test}+)}{P(\text{variant})} = 5.75$, hence $P(\text{variant}|\text{Test}+) \approx 14\%$.

Most positive results are actually false alarms. Another classical example is that of the prosecutor's fallacy: a small probability of evidence given innocence needs NOT mean a small probability of innocence given evidence.

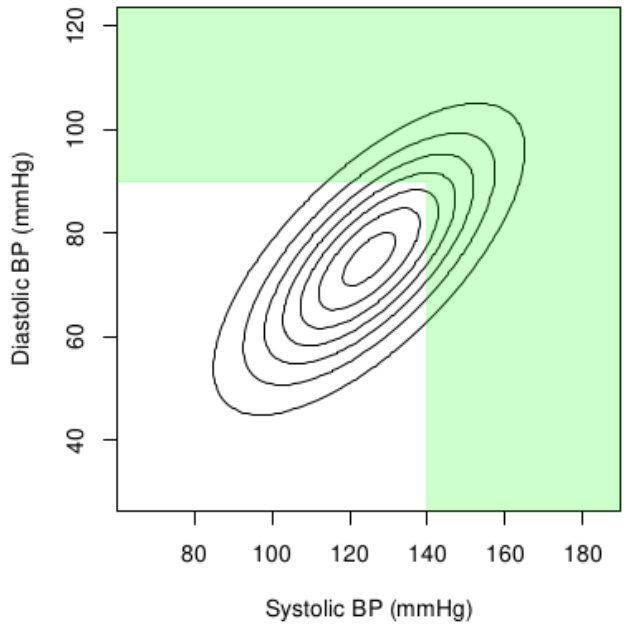
Bayes' Theorem

Continuous version

To get the probability of outcomes in a region, we integrate:



$$\mathbb{P}\left[\begin{array}{c} 100 < \text{SBP} < 140 \\ \& \\ 60 < \text{DBP} < 90 \end{array}\right] \approx 0.52$$



$$\mathbb{P}\left[\begin{array}{c} \text{SBP} > 140 \\ \text{OR} \\ \text{DBP} > 90 \end{array}\right] \approx 0.28$$

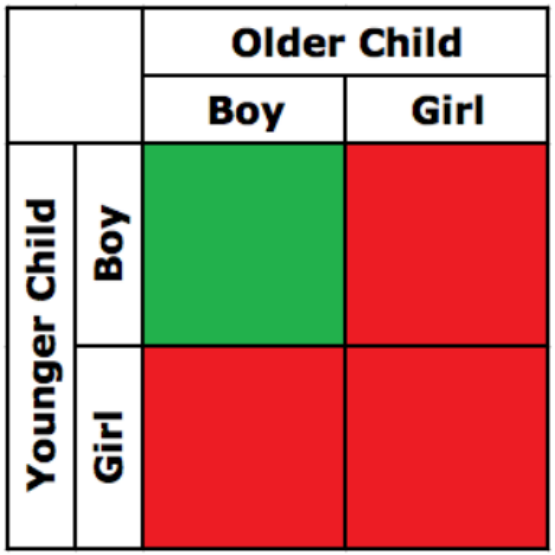
And Bayes' theorem is expressed ad relationships between conditional densities.

Bayes' theorem

Some "genetics"

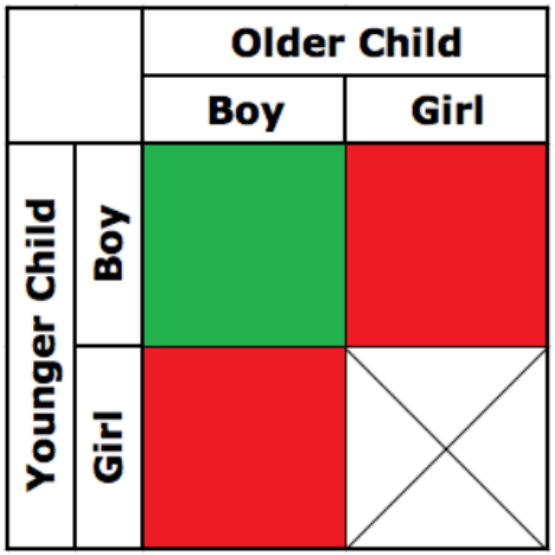
A randomly-chosen father of two has two kids. Given that at least one is a boy; what is the probability he has two boys?

Unconditional



$\mathbb{P}[2 \text{ Boys}] = 1/4 = 0.25$

Conditional



$\mathbb{P}[2 \text{ Boys} | 1 \text{ Boy}] = 1/3 \approx 0.33$

Bayes' theorem

Some "genetics"

Now a problem – not a trick! – to show that conditional probability can be non-intuitive, and careful reasoning is needed.

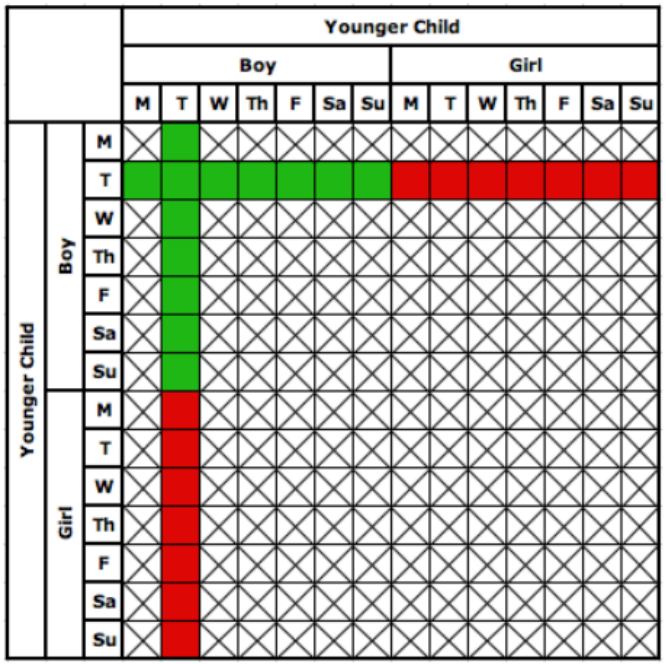
In the same setting, given that at least one is a boy who was born on a Tuesday ; what is the probability the father has two boys?

- The 'obvious' (but wrong!) answer is to stick with 1/3. What can Tuesday possibly have to do with it?
- It may help your intuition, to note that a boy being born on a Tuesday is a (fairly) rare event
  - Having two sons would give Jo two chances of experiencing this rare event
  - Having only one would give him one chance
  - 'Conditioning' means we know this event occurred, i.e. Jo was "lucky" enough to have the event
- Easier question: Is $P(2 \text{ Boys} | 1 \text{ Tues Boy}) > 1/3$? or $< 1/3$?

Bayes' theorem

Some "genetics"

Conditioning on at least one Tuesday-born boy



This gives $P(2 \text{ Boys} | 1 \text{ Tues Boy}) = 13/27 \approx 0.48$, quite different from $1/3 \approx 0.33$

Conditional independence

An example

- Data - suppose we know events:
  $F = \{$ a patient develops cancer $\}$
  $G = \{$ patient's parent's genotype $\}$
  $H = \{$ patient's genotype $\}$
- Informal statement: if we know the patient's genotype H, does knowledge of the parents' genotype G give any additional information?
  Formal statement:does $P(F|H) = P(F|G, H)$?
- Answer: in general, conditional independence will hold, but not on all occasions; in genomic imprinting genes are expressed in a parent-of-origin-specific manner, i.e., the expression of the gene depends upon the parent who passed on the gene

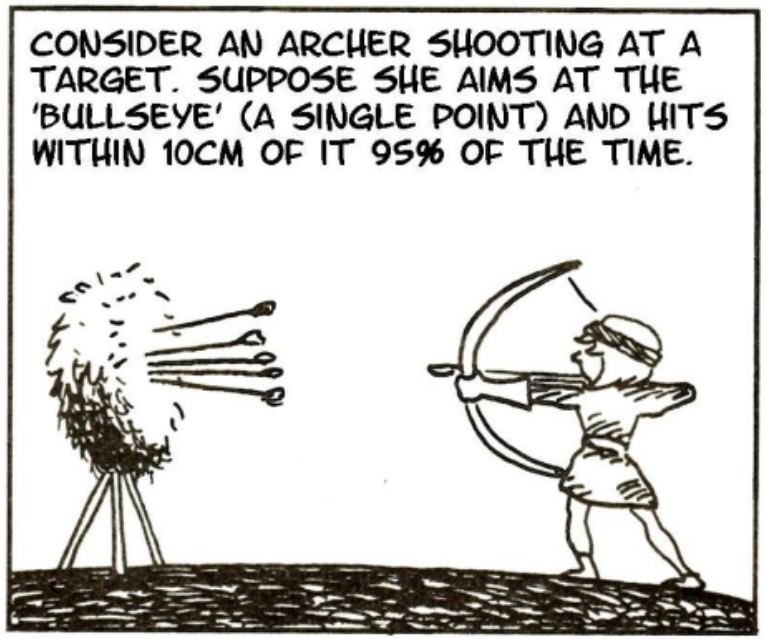
Bayesian Statistics

Nothing controversial so far, it was only math

In addition to describing random variables, Bayesian statistics uses the language of probability to describe what is known about unknown parameters.
Frequentist statistics , e.g. using p-values & confidence intervals, does not quantify what is known about parameters
Talk about Bayesian updating of knowledge. Arrow in target example or search for AF447 flight blackbox from wreck parts

Bayesian Inference

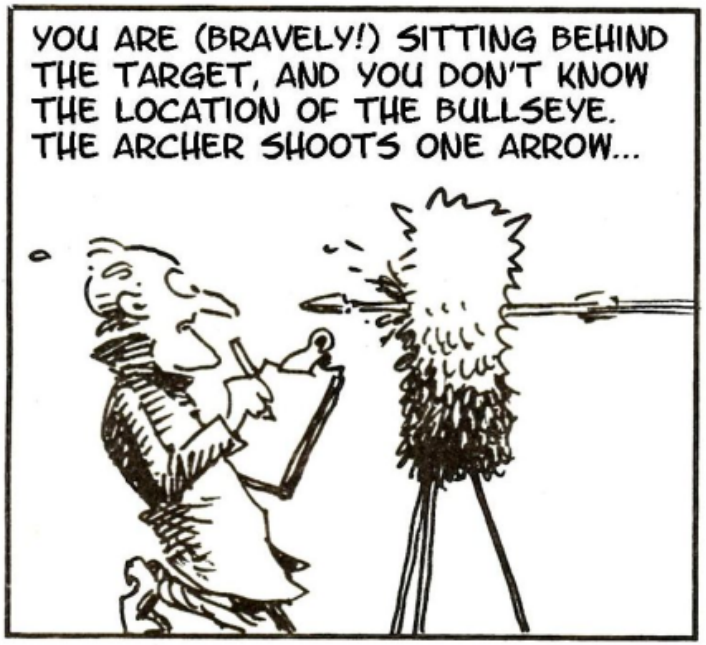
How it works



(adapted from Gonick & Smith, *The Cartoon Guide to Statistics*)

Bayesian Inference

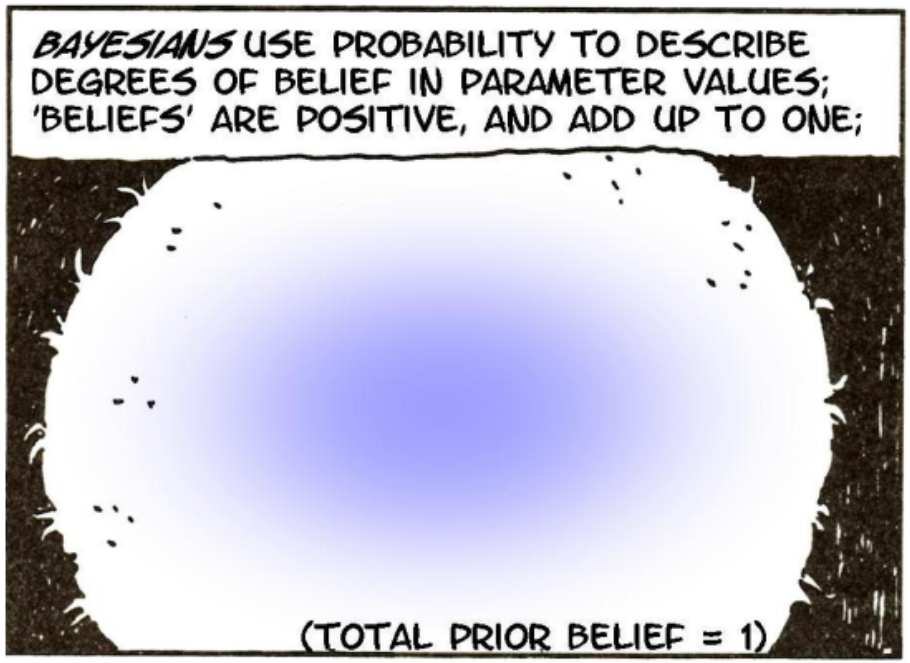
How it works



Bayesian Inference

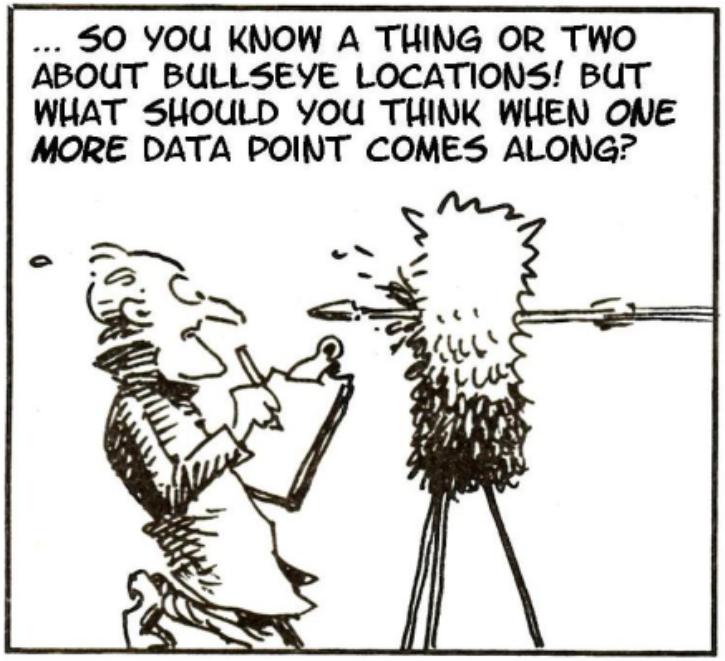
How it works

You don't know the location exactly, but do have some ideas.



Bayesian Inference

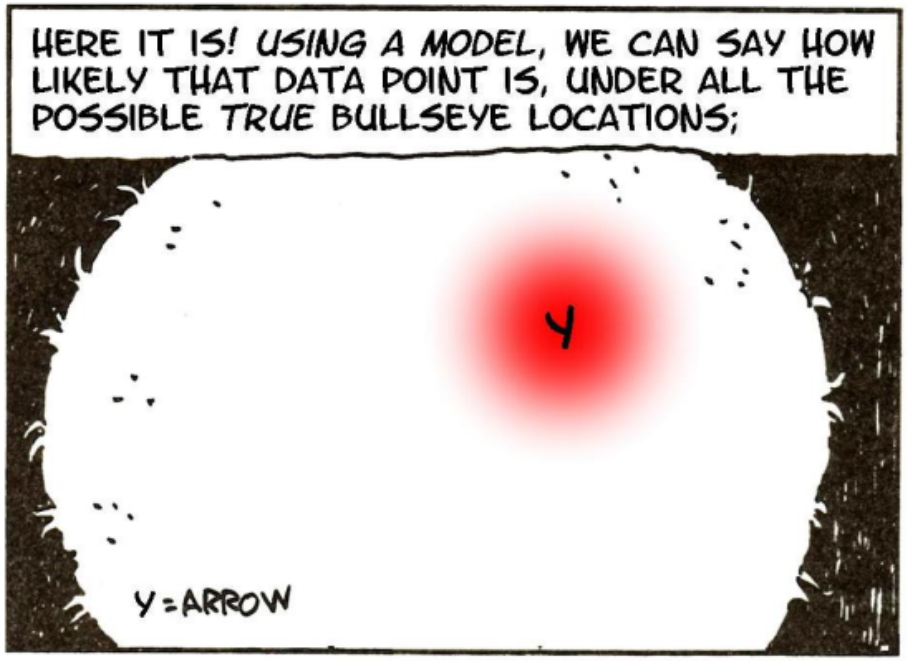
How it works



Bayesian Inference

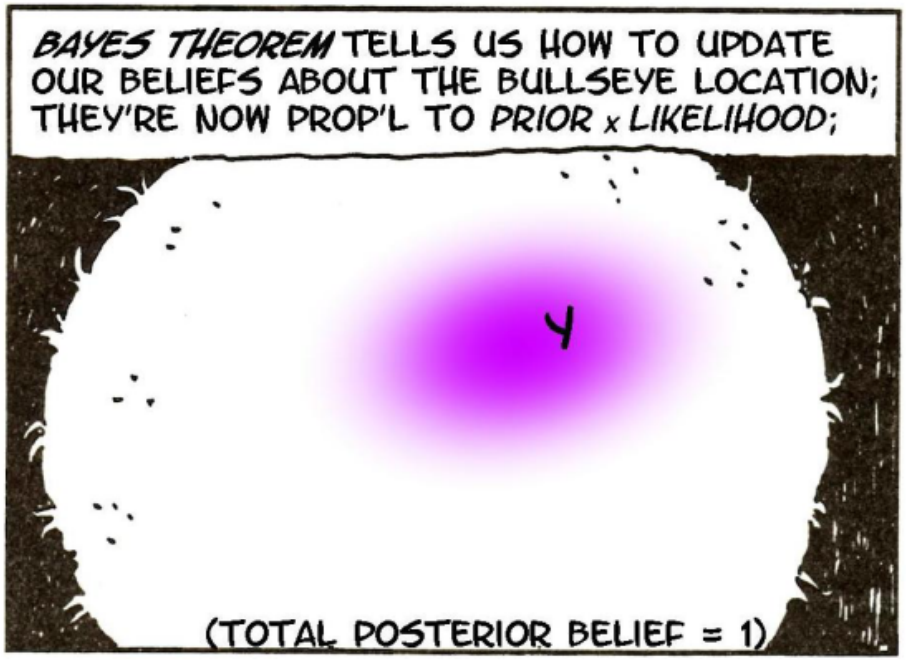
How it works

What to do when the data comes along?



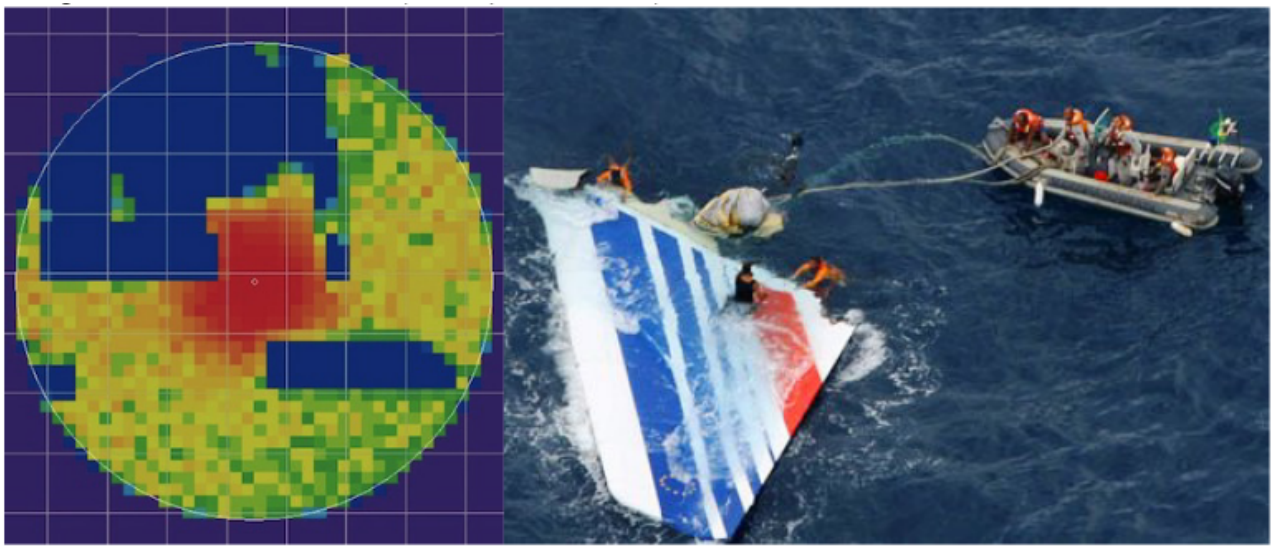
Bayesian Inference

How it works



Bayesian Inference

Exactly the same idea in practice



During the search for Air France 447, from 2009-2011, knowledge about the black box location was described via probability – i.e. using Bayesian inference

Eventually, the black box was found in the red area

Bayesian Inference

How to update knowledge as more data is collected?

- Prior distribution is what you know about your unknown parameter θ, without using the information in the data. Denoted $p(\theta)$
- Likelihood is the model for sampling and distribution assumption on how the data is produced. It says how likely the data is *if* the parameter value was truly θ: $p(y|\theta)$.

Bayes theorem allows you to computer the posterior distribution:

$$\begin{aligned} p(\theta|y) &\propto p(y|\theta) \times p(\theta) \\ \text{Posterior} &\propto \text{Likelihood} \times \text{Prior} \end{aligned}$$

Bayesian Inference

And that's it

Essentially.

- Given modelling assumptions and prior, the process is *automatic*
- Keep adding data and updating knowledge. As data will accumulate (if the model is correct), the knowledge will concentrate around the true θ
- It is a rational process with a specified prior to do so and update it, when given observed data.

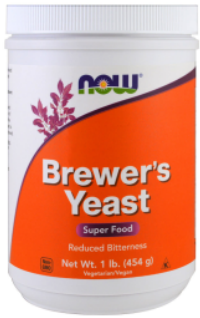
Hands-on time

- Find the file `BayesI.pdf` (Generate data from a simple genetic mixture model), or even better the `BayesI.Rmd` file, compile it, read and let's try to get to Exercise 1. Wait for Exercises 2 and 3 :)

Bayesian inference

ASE example

In an allele specific expression (ASE) experiment, two strains (BY and RM) are hybridised.

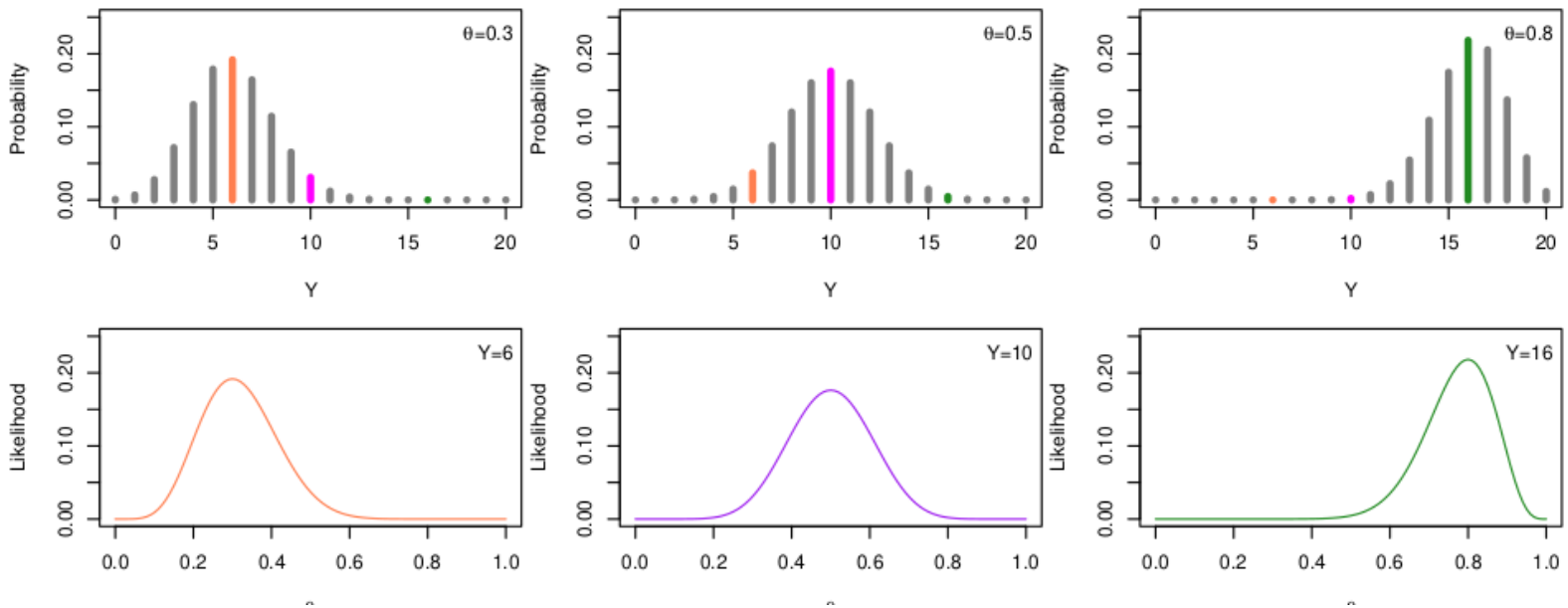


- N denotes the total number of expression reads at a particular location in the genome, Y denotes the number from BY
- We define θ as the probability a read come from BY (not RM)
- How far θ is from 0.5 determines how much ASE there is

Bayesian inference

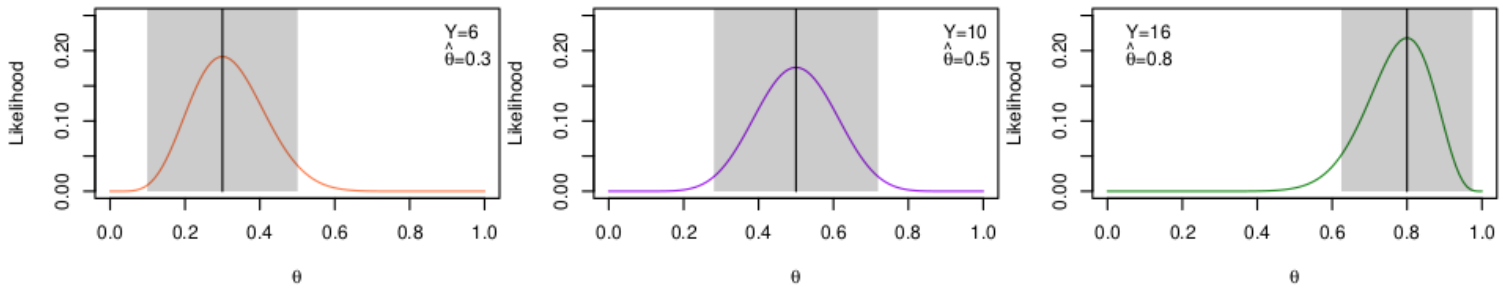
ASE example

Two ways to look at $P(y|\theta)$, varying y (1st row) or varying θ (2nd row)



What does classical analysis do?

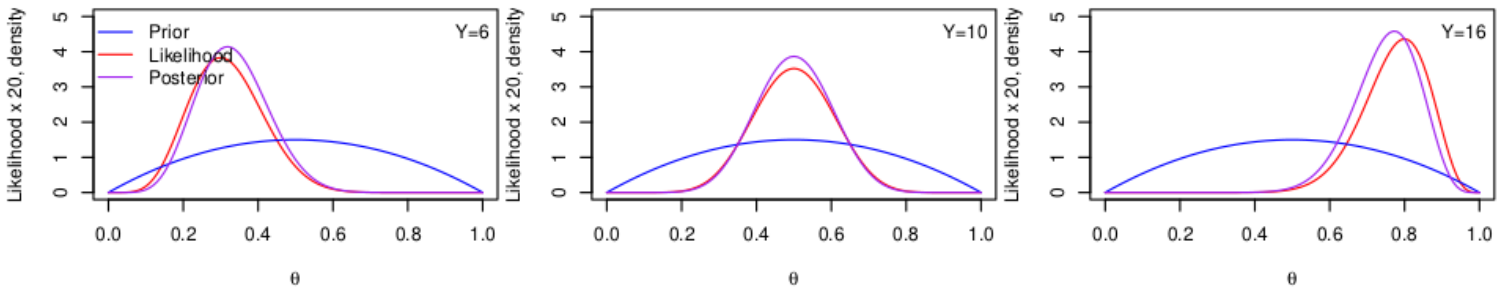
ASE example



- The point estimate (vertical line) is $\hat{\theta} = \hat{Y} = Y/N$, and an estimate of its standard error is given by $\sqrt{\hat{\theta}(1-\hat{\theta})/N}$.
- An approximate 95% CI (shaded region) is given by $\hat{\theta} \pm 1.96 \times s.e.$. This is an interval which, over many experiments, covers the true θ in (approximately) 95% of them
- The analysis doe not (& cannot) tell us if any given experiment's CI is in the 95% or the 5%.

Bayesian analysis

ASE example



- This prior gives most support near $\theta = 0.5$ (mild ASE) decreasing to 0 at $\theta = 0, 1$ (expression impossible/guaranteed in BY)
- The prior's influence is to make results slightly more conservative than using likelihood alone
- Formally, this is statistical induction: reasoning from specific data to general population characteristics
- Keen people: only relative size of likelihood & prior matters

How to summarise a posterior

The posterior distribution **is** the output of a Bayesian analysis. You may want to look at doing:

- Estimation marginal posterior distribution on parameters of interest. This would answer "test-like" questions. You need to integrate other all other parameters.
- Prediction via an output variable, aka the predictive distribution. You need to integrate the uncertainty of all parameters.

Reporting the entire posterior is too complex in most cases. One helpful summary is a *point estimate*. Our "best guess" for θ based on the posterior: posterior mean (centre of mass), posterior median (halfway point) or posterior mode (high point)

How to summarise a posterior

To summarise the posterior uncertainty, a natural analogue of the s.e. is the posterior standard deviation.

More directly, we can calculate *credible intervals* aka *compatibility intervals* aka *Bayesian confidence intervals*. For example, considering the 5% and 95% quantiles of the posterior distribution, one can build a 90% credible interval for that quantity. The interpretation is that the probability of the true parameter lies in this interval **is** 90% !!

Hands-on time

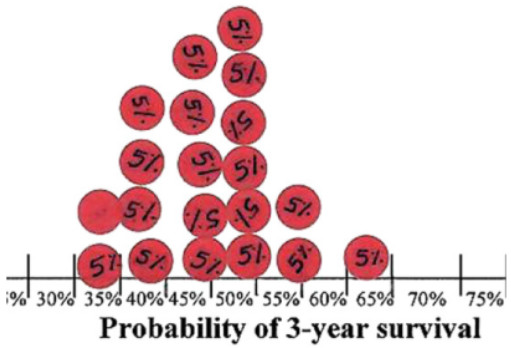
- Then read `BayesII.pdf` (Comparing two models with a likelihood ratio), `BayesIII.pdf` (Likelihood Ratio: how big is convincing?) and `BayesIV.pdf` (Bayesian calculation for comparing K models). Go back to Exercise 2 of `BayesI.pdf`.
- Then read `BayesV.pdf` (The Likelihood Function), `BayesVI.pdf` (Bayesian inference for a binomial proportion) and `BayesVII.pdf` (Conjugate Bayesian Analysis). And do Exercise 3 of `BayesI.pdf`.

Not so simple: where do priors come from?

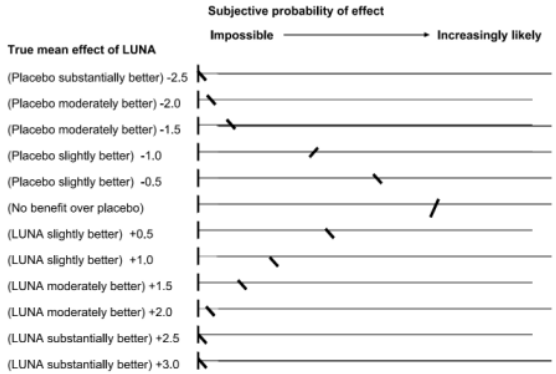
- Priors come from all data external to the current study. Everything else available.
- Boiling down what subject-matter experts know/think is known is called *eliciting a prior*
- Like eliciting effect sizes for power calculations, it is not easy (books exist on the topic!). Some tips:
  - Discuss parameters experts understand: code variables in familiar units, make comparisons relative to an easy-to-understand reference, not when age=height=IQ=0!
  - Avoid leading questions (just like in surveys)
  - The language of probability is unfamiliar, help users express their uncertainty (see Kynn JRSSA 2008 for a review).

Not so simple: where do priors come from?

Some ideas to help experts "translate" their knowledge to the language of probability:



Use 20×5% stickers (Johnson *et al* 2010, J Clin Epi) for prior on survival when taking warfarin

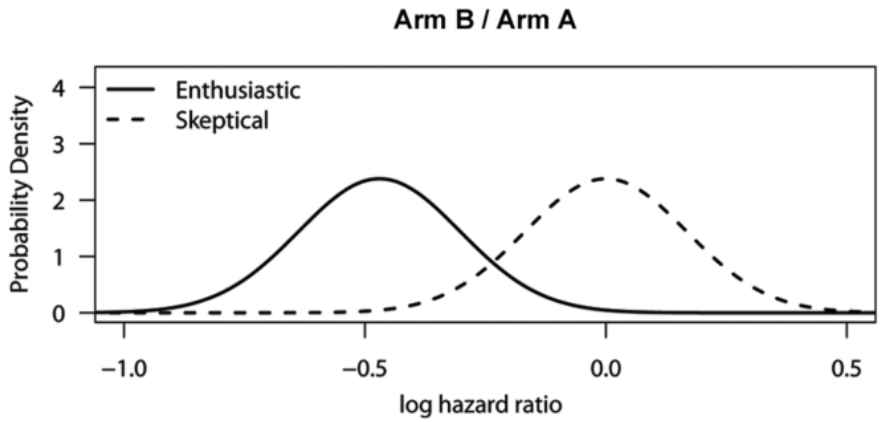


Normalize marks (Latthe *et al* 2005, J Obs Gync) for prior on pain effect of LUNA vs placebo

Typically, these "coarse" priors are smoothed. Providing the basic shape remains, exactly how much you smooth is unlikely to be critical in practice.

Not so simple: where do priors come from?

What to do if the experts disagree? Try it both ways.



I one had this prior and the data, this is the posterior one would have. If one had that prior... etc.
If the posteriors differ, what You believe based on the data depends, importantly, on Your prior knowledge. To convince other people expect to have to convince skeptics – and note that convincing [rational] skeptics is what science is all about.

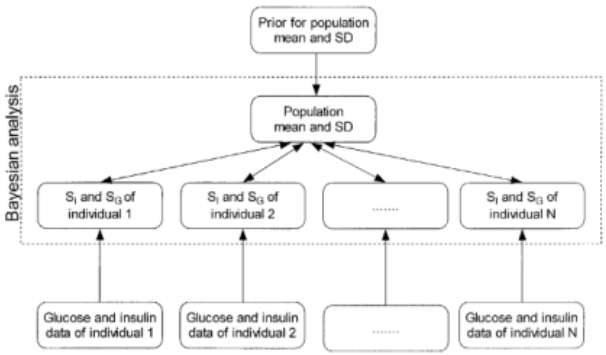
Sometimes priors don't matter

- Typically when the data provides a lot more information than the prior; the likelihood dominates.
- A word of caution to use flat prior to represent ignorance:
  - Flat priors actually give most of their support to extreme values, usually easily ruled out
  - If thinking about uniform priors, you cannot be uniform on all scales (parameter transformation) and on an infinite range $\rightarrow$ *improper* prior
  - However, in "famous" regression models, flat priors actually represent "ignorance" and classical and Bayesian outputs match. "Objective Bayes" will seek priors that are minimally informative (hard to define). Other names: reference, non-subjective.

Where are Bayesian methods commonly used?

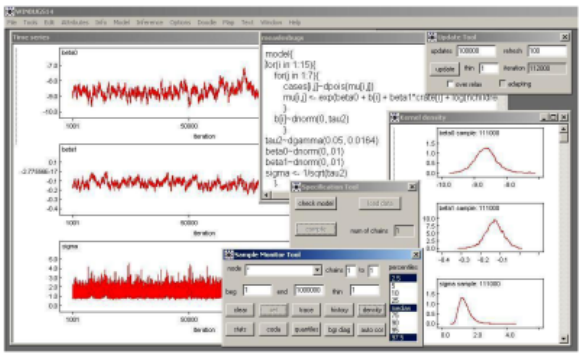
Allowing approximate Bayes, one answer is "almost in any field".
More explicitly:

Hierarchical modeling



One expert calls the classic frequentist version a "statistical no-man's land"

Complex models



...for e.g. messy data, measurement error, multiple sources of data; fitting them is *possible* under Bayesian approaches, but perhaps still not easy

Common Bayesian inference software

Pieces of software you can call from R (or Python, Julia, Matlab...):

- (Win/Open/Multi)BUGS, jags (declarative), stan (imperative), greta, R-INLA (formula-based)...
- They have slightly different technology
- Actually "Past, Present and Future of Software for Bayesian Inference", Štrumbelj et al., *Stat. Science* 2024. (took examples in there)

In "Evaluation of Bayesian alphabet and GBLUP based on different marker density for genomic prediction in Alpine Merino sheep", Zhu et al., *G3 Genes—Genomes—Genetics*, 2021.

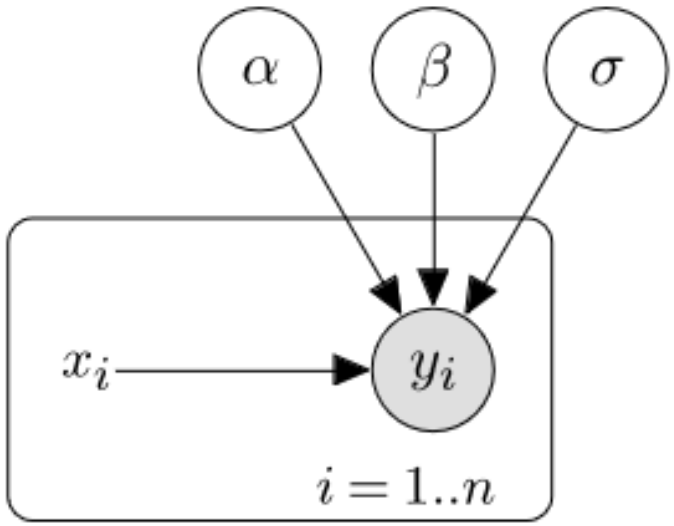
"The BayesA assumes that all SNPs have genetic effects and the variance of marker effects should obey the t-distribution, whereas BayesB assumes that only a small proportion of SNPs have an effect. Furthermore, the BayesCπ is similar to BayesB, and estimates the proportion of sites with no effect of π in the model. The Bayesian LASSO..."

Modelling language example

Bayesian linear regression in...

Mathematical terms

$$
\begin{aligned}
y_i|\beta, \alpha, \sigma, x_i &\sim \text{Normal}(\beta x_i + \alpha, \sigma^2),\ i = 1, \ldots, n, \\
\alpha &\sim \text{Normal}(0, 5\ 2), \\
\beta &\sim \text{Normal}(0, 5\ 2), \\
\sigma &\sim \text{Unif}(0, 10)
\end{aligned}
$$



Modelling language example

Bayesian linear regression in...

JAGS

```
model {
 for (i in 1:n) {
  y[i] ~ dnorm(beta*x[i] + alpha, 1 / (sigma*sigma))
 }
 alpha ~ dnorm(0, 1 / 25)
 beta ~ dnorm(0, 1 / 25)
 sigma ~ dunif(0, 10)
}
```

Modelling language example

Bayesian linear regression in...

PyMC

```
with Model() as model:
  sigma = Uniform("sigma", lower = 0, upper = 10)
  alpha = Normal("alpha", 0, sigma = 5)
  beta = Normal("beta", 0, sigma = 5)

  likelihood = Normal("y", mu = beta * x + alpha,
  sigma = sigma, observed = y)
```

Modelling language example

Bayesian linear regression in...

Stan

```
data {
  int<lower=0> n;
  vector[n] x;
  vector[n] y;
}
parameters {
  real alpha;
  real beta;
  real<lower=0, upper=10> sigma;
}
model {
  y ~ normal(beta * x + alpha, sigma);
  alpha ~ normal(0, 5);
  beta ~ normal(0, 5);
}
```

Modelling language example

Bayesian linear regression in...

Bean Machine

```
@bm.random_variable
def alpha():
  return Normal(0, 5)
@bm.random_variable
def beta():
  return Normal(0, 5)
@bm.random_variable
def sigma():
  return Uniform(0, 1)
@bm.random_variable
def x(i):
  return Normal(0, sigma())
@bm.random_variable
def y():
  return Normal(logit = beta() * x + alpha(), sigma())
```

Modelling language example

Bayesian linear regression in...

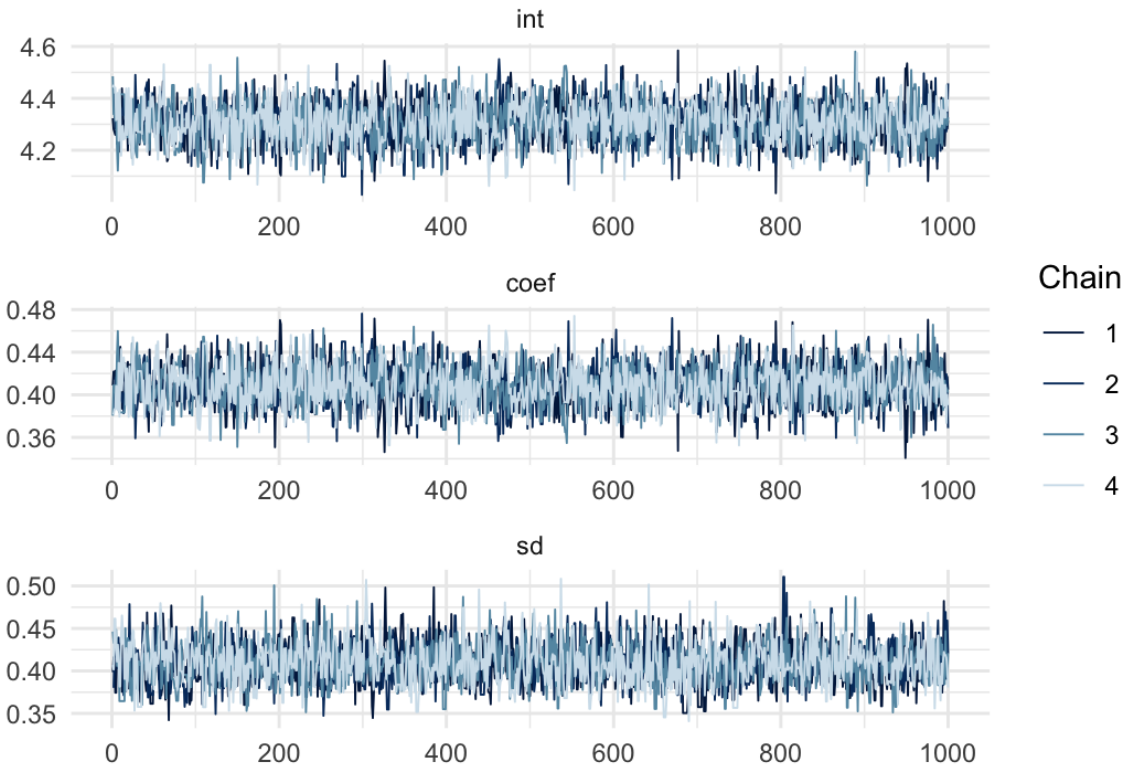
Greta

```
alpha <- normal(0, 5)
beta <- normal(0, 5)
sigma <- uniform(0, 10)

distribution(y) <- normal(beta * x + alpha, sigma)
```

Basic Greta regression example - Iris data

```
#data
x <- as_data(iris$Petal.Length)
y <- as_data(iris$Sepal.Length)

library(greta)
#variables and priors
int <- normal(0, 1)
coef <- normal(0, 3)
sd <- lognormal(0, 3)

mean <- int + coef * x # an operation
distribution(y) <- normal(mean, sd) #likelihood

m <- model(int, coef, sd) # model definition

draws <- mcmc(m, n_samples = 1000, chains = 4) # sampling
bayesplot::mcmc_trace(draws)
```

Basic Greta regression example - Iris data



Monte Carlo methods

Monte Carlo methods

General idea

Definition

Monte Carlo is a class of methods that excels in the art of approximating an expectation by the sample mean of a function of simulated random variables.

- Spoiler alert: Monte Carlo will heavily rely on the Law of Large Numbers to approximate expectations
- Expectations: with $X \sim p(x)$: $E[g(X)] = \int_{x \in \xi} g(x)p(x)dx$ in the continuous case or $E[g(X)] = \sum_{x \in \xi} g(x)p(x)$.
- We have seen and we will see many cases as to why this is a crucial problem in Bayesian inference.

Monte Carlo methods

A discerete example

Imagine that s is the count of eggs laid by hens of a given species.
We are given the distribution:

s	0	1	2	3	4	5	6	7	8	9	10
p(s)	0.02	0.07	0.15	0.20	0.20	0.16	0.10	0.06	0.03	0.01	0.01

The expected family size is $E[S] = \sum_{i=0}^{10} s.p(s)$.
But if you are interested in the number of sibling pairs in a family.
With s offspring, there are $g(s) = s(s-1)/2$ sibling pairs:

$$E[g(S)] = \sum_{i=0}^{10} [s(s-1)/2]\, p(s).$$

Important: for every Monte Carle approximation, there are (a) an underlying distribution and (b) a function $g(x)$. Identify them!

Monte Carlo methods

Law of large numbers

Theorem

Weak Law of Large Numbers Let $X_1, X_2, \ldots, X_n$ be iid r.v.'s with $E|Xi| < \infty$, and $\bar{X}_n = \sum_{i=1}^{n} X_i$. Then:

$$\lim_{n \to \infty} P(|\bar{X}_n - E[X_i]| > \epsilon) = 0 \text{ for any } \epsilon > 0$$

Translation: if you take a sample mean of random variables, as your sample size gets very large, the sample mean gets very close to the expectation.

This suggests: an expectation might be approximated by the sample mean of n random variables (and it will work better if n is large).

$$E[g(X)] \approx \frac{a}{n} \sum_{i=1}^{n} g(x_i), \text{ with } x_i \sim p(x).$$

Monte Carlo methods

Why estimating the expectation is useful?

Usually, any quantity of interest may be expressed as the expected value of a function of some random variable. Importantly:

- $$P(X \in \mathcal{A}) = E[I_{\mathcal{A}}(X)],$$

  with $I_{\mathcal{A}}(x)$ the indicator function of set $\mathcal{A}$, taking value 1 if $x \in \mathcal{A}$ and 0 otherwise.

- $$\int_a^b q(x)dx = (b-a)E\left[q(U)\right],$$

  with $U \sim \text{Unif}[a, b)$.

Monte Carlo methods

Variance of Monte Carlo estimators

A Monte Carlo estimator is simply a random variable itself—a sum of random variables:

$$G_n = \frac{1}{n}\sum_{i=1}^{n} g(X_i).$$

So if the X_i are independent (this will not be the case for MCMC; this is only to give you an idea), the variance of G_n can be easily computed:

$$\mathrm{Var}(G_n) = \frac{\mathrm{Var}(g(X_i))}{n}$$

.

So we can reduce this variance by (i) increasing the number of samples n and/or (ii) reducing $\mathrm{Var}(g(X_i))$.

Monte Carlo methods

Sampling from a Beta (posterior)

Go to `BayesVIII.pdf` (Monte Carlo sampling from a Beta Posterior distribution)

Markov Chains

Markov Chains

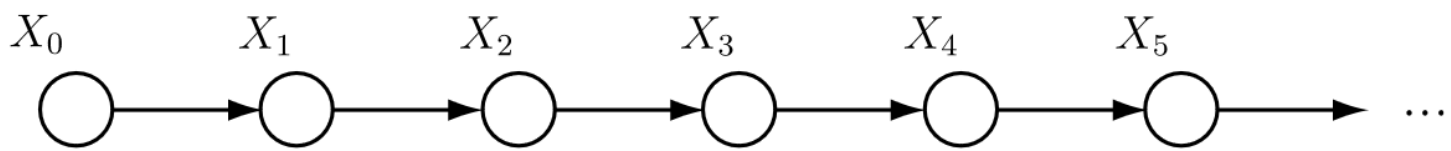
Introducing Markov chains

Definition

X_t , $t = 0, 1, 2 \ldots$, having a joint distribution such that:

$$P(X_t|X_{t-1}, X_{t-2}, \ldots, X_0) = P(X_t|X_{t-1}), \forall t$$

Graphically:



X_t are mainly scalars for us (one parameter), but could be multi-(or even high-)dimensional in an MCMC application.

Markov Chains

A genetics example: the Wright-Fisher population

Let X_t denote the number of alleles of type 'A' in a Wright-Fisher population of size N diploids. Then:

$$P(X_t = j | X_{t-1} = i) = \frac{(2N)!}{(2N-j)!j!} \left(\frac{i}{2N}\right)^j \left(\frac{2N-i}{2N}\right)^{2N-j}$$

- Important point: conditional independence $\neq$ independence. If you don't know X_{t-1}, then, of course X_t depends on X_{t-2}.
- NB: Do not think that MCMC is useful in genetics because the problems in genetics involve Markov chains. The Markov chain underlying Markov chain Monte Carlo and any Markov chains involved in a statistical genetics model may be quite distinct entities.

Markov chains

Transition probability matrix

We can write down the transition probabilities from all values of X_t to all values of X_{t+1} in a matrix ($P(i \to j)$ as a shorthand for $P(X_{t+1} = j | X_t = i)$):

$$
\begin{array}{c}
\nearrow \\
0 \\
1 \\
2 \\
3 \\
\vdots \\
2N
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 0 & & 1 & & 2 & \cdots & 2N \end{array} \\
\begin{pmatrix}
P(0 \to 0) & P(0 \to 1) & P(0 \to 2) & \cdots & P(0 \to 2N) \\
P(1 \to 0) & P(1 \to 1) & P(1 \to 2) & \cdots & P(1 \to 2N) \\
P(2 \to 0) & P(2 \to 1) & P(2 \to 2) & \cdots & P(1 \to 2N) \\
P(3 \to 0) & P(3 \to 1) & P(3 \to 2) & \cdots & P(3 \to 2N) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P(2N \to 0) & P(2N \to 1) & P(2N \to 2) & \cdots & P(2N \to 2N)
\end{pmatrix}
\end{array}
$$

TPMs form the basis for much of the classical analysis of Markov chains.
Let's have a look at a simpler example.

Markov chains

Random walk with scattering boundaries

Imagine a random walk on the integers from 1 to 5. From 1 or 5, the walk goes to any state with equal probability, and from the remaining states, the walk may take steps of size 0 or 1, but they are biased toward the center.

For example, consider the TPM:

$$\mathbf{P} = \begin{array}{c} \nearrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} .2 & .2 & .2 & .2 & .2 \\ .2 & .3 & .5 & 0 & 0 \\ 0 & .3 & .4 & .3 & 0 \\ 0 & 0 & .5 & .3 & .2 \\ .2 & .2 & .2 & .2 & .2 \end{pmatrix} \end{array}$$

Computer example: `BayesIX.html` (Markov chain bouncing blob)

Markov chains

TPM, some properties

- The rows of P sum to one; they are probabilities that must sum to one.
- The columns need not to sum to one.
- Probabilities after one step can be computed by matrix multiplication. i.e., if: $v_0 = (0, 1, 0, 0, 0)$, (v's are row-vectors; the initial state is state 2), then the probabilities of being in states 1,2,. . . ,5 after one step of the chain are given by: $v_0.P = v_1 = (0.2, 0.3, 0.5, 0, 0)$
- Probabilities after two steps are: $v_1.P = (v_0.P).P = v_0.P^2 = v_2$
- and probabilities after n steps are:

$$v_n = v_0.P^n$$

Markov chains

Limiting distribution

A class of Markov chains called ergodic Markov chains have the property that a limiting distribution π exists s.t. (s states):

$$\lim_{n\to\infty} P^n = \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_s \\ \dots & \dots & \dots & \dots \\ \pi_1 & \pi_2 & \dots & \pi_s \end{pmatrix}$$

Take our random walk example

$$\mathbf{P}^1 = \begin{array}{c} \nearrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} .2 & .2 & .2 & .2 & .2 \\ .2 & .3 & .5 & 0 & 0 \\ 0 & .3 & .4 & .3 & 0 \\ 0 & 0 & .5 & .3 & .2 \\ .2 & .2 & .2 & .2 & .2 \end{pmatrix} \end{array}, \mathbf{P}^2 = \begin{array}{c} \nearrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0.12 & 0.20 & 0.36 & 0.20 & 0.12 \\ 0.10 & 0.28 & 0.39 & 0.19 & 0.04 \\ 0.06 & 0.21 & 0.46 & 0.21 & 0.06 \\ 0.04 & 0.19 & 0.39 & 0.28 & 0.10 \\ 0.12 & 0.20 & 0.36 & 0.20 & 0.12 \end{pmatrix} \end{array}$$

Markov chains

Limiting distribution

$$\mathbf{P}^3 = \begin{array}{c} \nearrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0.088 & 0.216 & 0.392 & 0.216 & 0.088 \\ 0.084 & 0.229 & 0.419 & 0.202 & 0.066 \\ 0.066 & 0.225 & 0.418 & 0.225 & 0.066 \\ 0.066 & 0.202 & 0.419 & 0.229 & 0.084 \\ 0.088 & 0.216 & 0.392 & 0.216 & 0.088 \end{pmatrix} \end{array}$$

$$\mathbf{P}^5 = \begin{array}{c} \nearrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0.075 & 0.219 & 0.412 & 0.219 & 0.075 \\ 0.074 & 0.220 & 0.415 & 0.218 & 0.073 \\ 0.072 & 0.220 & 0.415 & 0.220 & 0.072 \\ 0.073 & 0.218 & 0.415 & 0.220 & 0.074 \\ 0.075 & 0.219 & 0.412 & 0.219 & 0.075 \end{pmatrix} \end{array}$$

$$\mathbf{P}^\infty = \begin{array}{c} \nearrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0.07317 & 0.2195 & 0.4146 & 0.2195 & 0.07317 \\ 0.07317 & 0.2195 & 0.4146 & 0.2195 & 0.07317 \\ 0.07317 & 0.2195 & 0.4146 & 0.2195 & 0.07317 \\ 0.07317 & 0.2195 & 0.4146 & 0.2195 & 0.07317 \\ 0.07317 & 0.2195 & 0.4146 & 0.2195 & 0.07317 \end{pmatrix} \end{array}$$

Markov chains

Limiting distribution

- The limiting distribution in our rw example is:

$$\pi = (0.07317, 0.2195, 0.4146, 0.2195, 0.07317)$$

- If you start the chain from any of the five states, after a sufficient (and not very many in this case) number of steps, the probability that it will be found in any of the five states is essentially independent of its starting state.
- Thus, the states visited by an ergodic Markov chain may be used to compute a Monte Carlo average. That is MCMC.

Use of Markov chains in Monte Carlo

Monte Carlo with a Markov chain is pretty much the same as before:

$$E[g(X)] \approx \frac{1}{n} \sum_{i=1}^{n} g(x^{(i)}),$$

except that now, the $x^{(i)}$ are states visited by a Markov chain having a limiting distribution that we wish to sample from.
To implement this we must be able to construct an appropriate Markov chain. It must:

- be ergodic
- have the right limiting distribution (the posterior distribution!)

We will explain how that is done using the *Metropolis-Hastings* algorithm.

MCMC?

A few points to consider

- If you can simulate independent samples, $x^{(1)}, \ldots, x^{(n)}$, then, by all means, do so, and avoid MCMC if you can.
- MCMC is most useful when the desired distribution to be sampled from is "known only up to scale"—this means that the "shape" of the distribution is known but its normalising constant is not. Classical example :)

$$P(\theta|X) = \frac{P(X|\theta)P(X)}{\int_{\theta'} P(X|\theta')P(X)d\theta'}$$

as the denominator is a constant wrt θ. The numerator is the joint density of the data X and the parameters θ, which is often easy to compute. This is why MCMC is so useful to Bayesians.

Back to Markov chains

Conditions ensuring a chain is ergodic

A Markov chain is ergodic if:

1. It has *no transient states*, i.e., there are no states with an expected time to recurrence of ∞. (This is not an issue with a finite state space)
2. It is *irreducible*, i.e., any state in the state space is reachable from any other state in the state space in a finite number of steps.
3. It is *aperiodic*, i.e., there exists no pair of states i and j such that the probability of reaching j from i is non-zero only if the number of steps is an integer multiple of some period τ.

A first step into MCMC

Balance equation

- To construct an ergodic Markov chain with limiting distribution π, one possibility is via a theorem which tells us about a property of π

Theorem

If π is the limiting distribution of the ergodic Markov chain with TPM P, then π is the unique stationary distribution that satisfies the general balance equation

$$\pi.P = \pi$$

- So, if we can find P such that its unique stationary distribution is π, and π is the distribution we wish to sample from, then we can use the chain defined by P in MCMC.

A first step into MCMC

Balance equation (continued)

- This is, however, not a tractable approach in any problem of consequence. In general it will be harder to (or just as difficult to) solve the general balance equation for π as it will be to draw independent samples from π.
- A key to the problem is that all possible states in the distribution π must be considered simultaneously. In most interesting problems, that number of states can be astronomically huge.
- The solution to this conundrum is to not try to solve the general balance equation directly, but, instead satisfy a "locally-defined" balance condition (so that only two states in the state space need be considered simultaneously, rather than all of the states, simultaneously), AND do this in a way that ensures that general balance is satisfied.
- This a clever "trick" :)

A first step into MCMC

Time-reversible Markov chains and detailed balance

- A special class of Markov chains are called "time-reversible" Markov chains, because if you were to watch them running "backward in time" they would look just the same as the chain running "forward in time".
- The salient feature of such chains is that they satisfy the detailed balance (also called the local balance) condition.
- Detailed balance with respect to π between a pair of states i and j, is satisfied by a Markov chain with a stationary distribution π, and a TPM P having elements $P(x \to y)$ if

$$\pi(i)P(i \to j) = \pi(j)P(j \to i)$$

- It is easy to show (sum over all states i) that if detailed balance holds for every pair of states, then general balance is also satisfied.

A first step into MCMC

The Metropolis-Hastings Algorithm

The M-H algorithm provides a way to perform steps in a Markov chain that satisfy detailed balance.

Imagine you wish to simulate a dependent sample from a target distribution f , and you are currently in state i. The recipe is:

1. Propose changing the state from state i to a new state j. Draw the state j from a proposal distribution, $q(j|i)$ which may be conditional on i.
2. Accept and move to the new state j with probability R equal to the lesser of 1 or the Hastings ratio:

$$R(i \to j) = \min\left[1, \frac{f(j)}{f(i)} \times \frac{q(i|j)}{q(j|i)}\right]$$

   If you do not accept the move to state j, then stay in state i (where you are).
3. Iterate until *convergence*

A first Metropolis-Hastings example

Let θ be a variable with probability density function
$\theta \sim \text{Beta}(74, 128)$

Note that this is the posterior distribution of the frequency θ of allele A at a locus, given a uniform prior, and a sample of 100 diploids in which are found 73 copies of allele A. $f(\theta) \propto \theta^{73}(1-\theta)^{127}$

This distribution is known exactly, but for illustration, we will construct a Markov chain that has limiting distribution f, and sample from it.

A first Metropolis-Hastings example

Let θ be a variable with probability density function
$\theta \sim \text{Beta}(74, 128)$

Note that this is the posterior distribution of the frequency θ of allele A at a locus, given a uniform prior, and a sample of 100 diploids in which are found 73 copies of allele A. $f(\theta) \propto \theta^{73}(1-\theta)^{127}$

This distribution is known exactly, but for illustration, we will construct a Markov chain that has limiting distribution f, and sample from it.

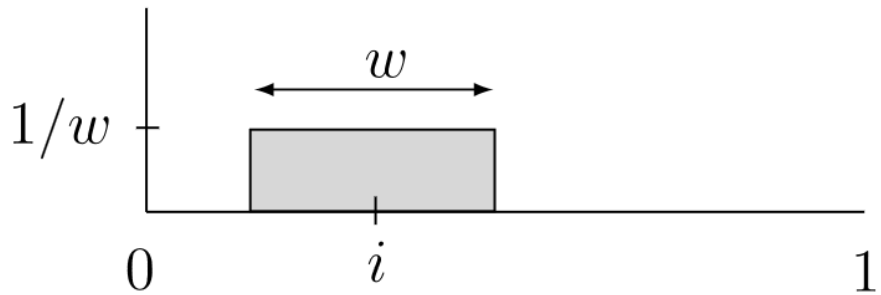
Thx for not complaining :). We are dealing with a continuous state space. The π, P, q and f encountered before are taken as probability density functions and the general balance equation is expressed by a collection of integrals (not matrix multiplication).

A first Metropolis-Hastings example

Step 1 - Choose a proposal distribution

One is free to choose any proposal distribution q. The only property that it should satisfy is that if $q(i|j) > 0$, then $q(j|i)$ should also be positive (and irreducibility etc. Clearly some proposals are better than others). Otherwise you end up wasting some time.

We will choose a uniform density for q with width w, centred on the current state i. It looks like



Thus, $q(i|j) = 1/w$ for all i and j. (Note that if $j \geq 1$ or $j \leq 0$ then $f(j) = 0$ and we reject the proposal immediately.)

A first Metropolis-Hastings example

Step two - Compute the Hastings Ratio

$$\frac{f(j)}{f(i)} \times \frac{q(i|j)}{q(j|i)} = \left(\frac{j}{i}\right)^{73} \left(\frac{1-j}{1-i}\right)^{127}$$

Notice that the normalising constant cancels out. (That always happens). And, in this case, q cancels out (only because q is symmetrical).

Generalising remarks on MCMC

- MCMC is just Monte Carlo with samples drawn from a Markov chain
- The Markov chain in MCMC is constructed by concatenating moves together, each of which satisfies detailed balance
- We will (try to) explore methods for implementing MCMC in problems relevant to statistical genetics

Computer examples: `BayesX.pdf` (The Metropolis Hastings Algorithm) and `BayesXI.pdf` (Simple Examples of Metropolis–Hastings Algorithm)

The Metropolis-Hastings algorithm in more than one dimension

Genotype Frequencies and Inbreeding

We work here with one locus with two alleles, A and a, at frequencies p and $1-p$, respectively, and *inbreeding coefficient* f.
The probabilities of the three genotypes are:

$$P(AA) = fp + (1-f)p^2$$

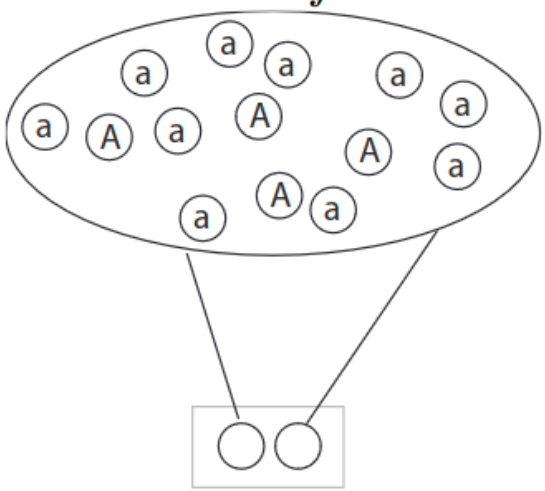
$$P(Aa \text{ or } aA) = (1-f)2p(1-p)$$

$$P(aa) = f(1-p) + (1-f)(1-p)^2$$

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

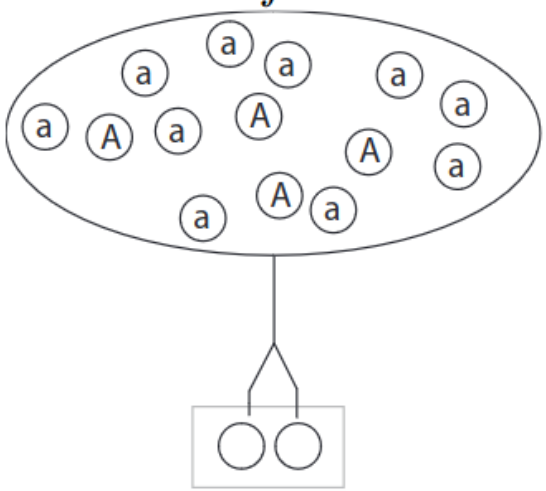
Inbreeding model

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



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

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

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: $n = 50$ with:

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

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 $f \sim \mathrm{Beta}(1,1)$ and $p \sim \mathrm{Beta}(1,1)$
- 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

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

Bayesian Model for Estimating f and p:

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:

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

as long as $0 < p < 1$ and $0 < f < 1$, otherwise it's 0.

Two-Dimensional Metropolis-Hastings 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.

For p we will choose a normal proposal distribution centred on the current value with standard deviation of s_p:

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

And, we will use the same for f

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

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

2D joint Metropolis-Hastings Sampler

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

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

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

   If you reject the proposed values, leave the current values unchanged.

MultiD Metropolis-Hastings algorithm - Remarks

- In MCMC, the proposal distribution need not propose changes to every variable/parameter in the model.
- Actually, few real-world problems require MCMC for which you would use a single proposal distribution to propose changes to all the variables in the model.
  - Any proposal distribution -regardless of how many/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 M-H algorithm).
  - These different flavours 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. Though some are "better" as they will lead to better convergence/mixing.

2D single component-wise Metropolis-Hastings Sampler for p and f

Simplest scenario has a sweep as follows:

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

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

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

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

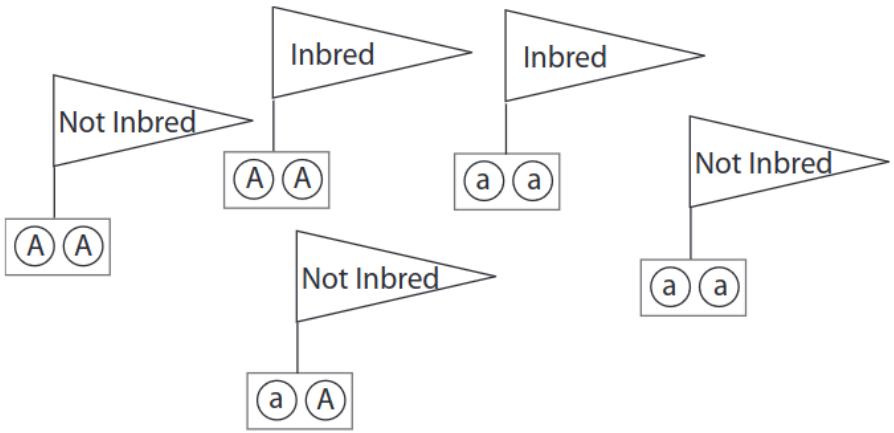
2D single component-wise Metropolis-Hastings Sampler for p and f

- The proposals are each a little simpler (slightly) than jointly proposing changes to (p, f). 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.
- You need both steps 1. and 2. from the previous algorithm to create an irreducible chain. If you never update p, for example, your chain could never reach every possible value of 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 "clever.

This will make us think about Gibbs sampling.
Computer practical: you can work on `inreeding-model-mcmc`

Formulating the Model with Latent Variables

Imagine how easy it would be to estimate f and p if we knew whether every individual we sampled was inbred or not

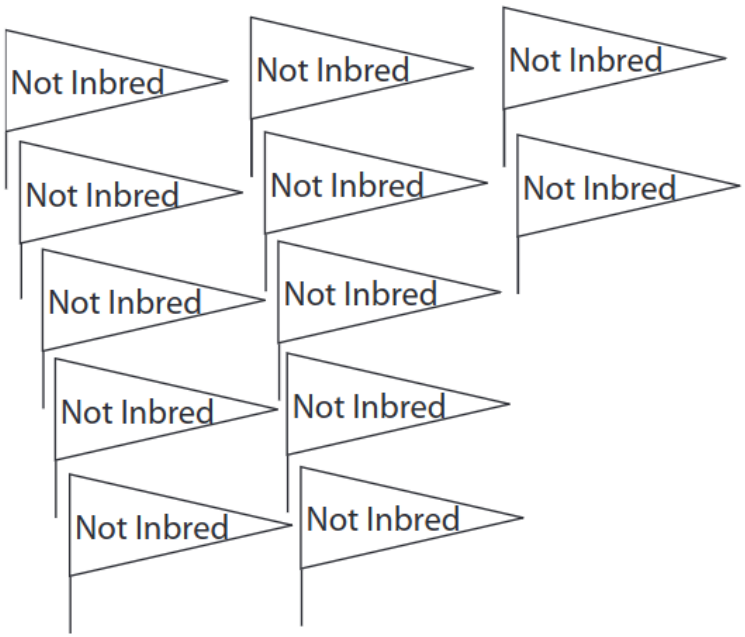


- f is just a binomial proportion
- To estimate p you count both the alleles from Non-inbred individuals, and just one allele from Inbred individuals. Then it too is simply a binomial proportion.

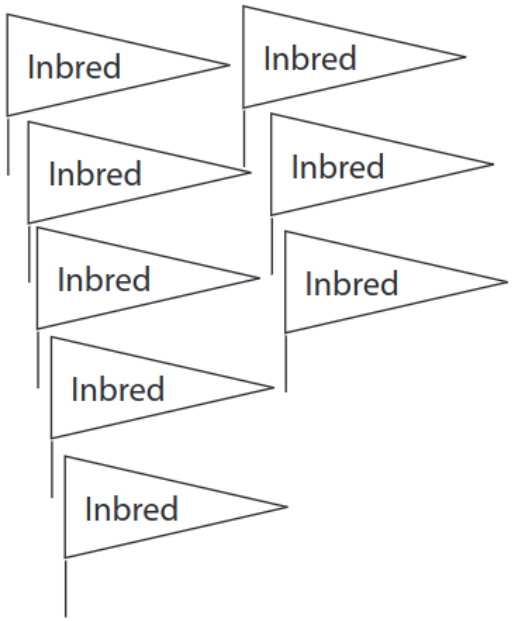
Formulating the Model with Latent Variables

To estimate f, you can forget the alleles carried by anyone.

You can just count the flags



12

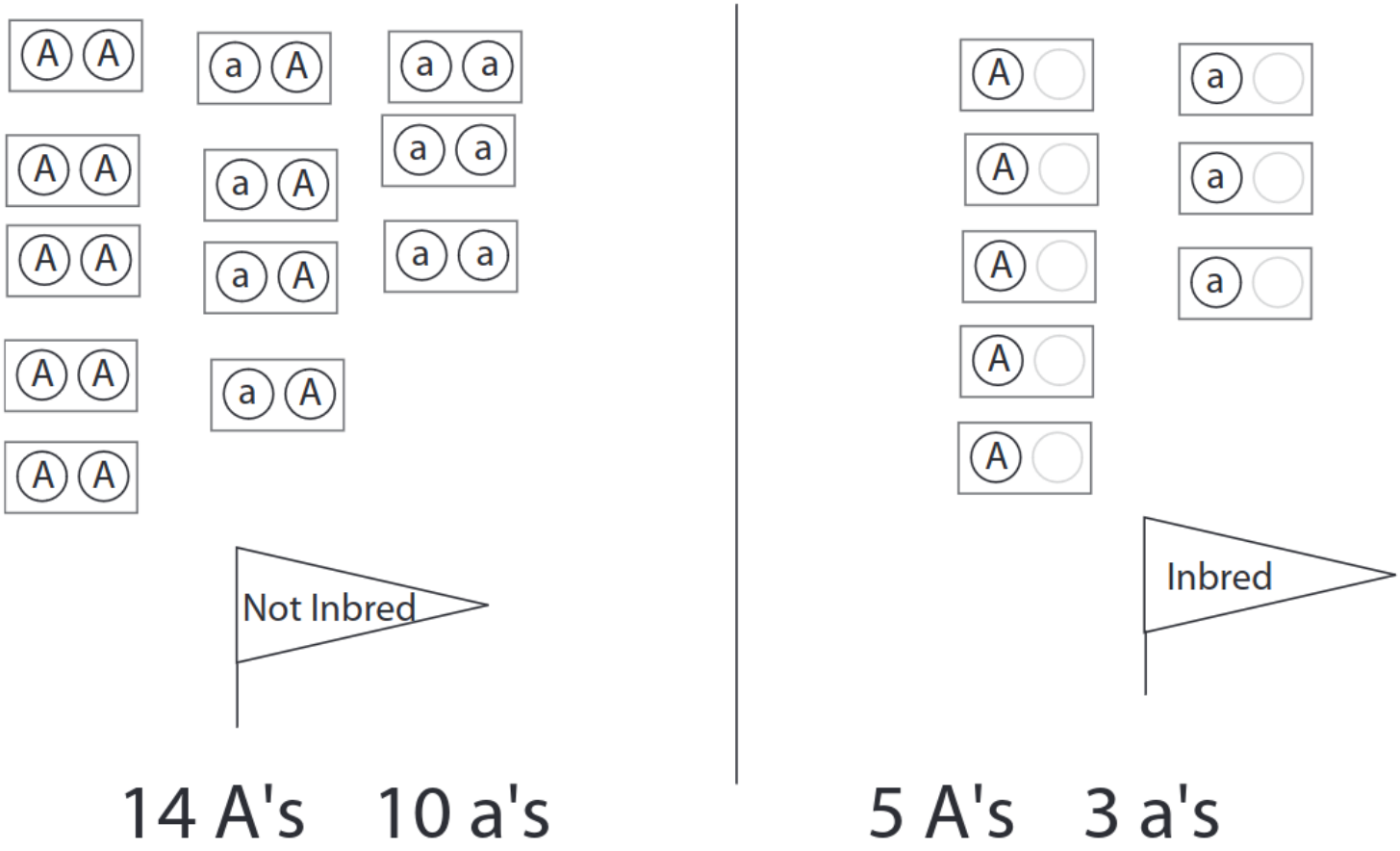


8

Formulating the Model with Latent Variables

To estimate p you just have to count alleles

Each inbred individual contributes just a single gene copy.



Joint Density of p, f and U

- We denote the n latent variables as $U = (U_1, ..., U_n)$.
- $u_i = 0$ indicates that i is Not Inbred and $u_i = 1$ indicates i is Inbred.
- Let $Y = (Y_1, ..., Y_n)$ denote the genotypes

The joint density is

$$P(p, f, U, Y) = P(p)P(f)P(U|f)P(Y|U, p)$$

Hence the posterior is

$$P(p, f, U|Y) = \frac{P(p, f, U, Y)}{\int_p \int_f \sum_{0 \leq U_1 \leq 1} \cdots \sum_{0 \leq U_n \leq 1} P(p, f, U, Y).dp.df}$$

The normalizing constant is nasty! We do not have to mess up with it. But what have we gained?

Generic MCMC for f and p with latent variables

- Recall, we wish to simulate f and p from their posterior distributions and so learn about $P(f, p|Y)$.
- However, now, our chain is moving about the space of f, p and U
  - it visits a sequence of states of the form $(f^{(1)}, p^{(1)}, U^{(1)}), (f^{(2)}, p^{(2)}, U^{(2)}), (f^{(3)}, p^{(3)}, U^{(3)}), \ldots$
- Obtaining the "marginal" (f, p) posterior distribution from the MCMC output is simple
  - From the sequence $(f^{(1)}, p^{(1)}, U^{(1)}), (f^{(2)}, p^{(2)}, U^{(2)}), \ldots$, yo usimply discard the U's. Though you might be interested in them too.

Naive Metropolis-Hastings for f, p, and U

Pretty similar to the 2D single component-wise MH algorithm, with Hastings ratios:

1. for p:

$$\alpha = \frac{q(p|p^*)}{q(p^*|p)} \frac{P(p^*)P(f)P(U|f)P(Y|U,p^*)}{P(p)P(f)P(U|f)P(Y|U,p)}$$

2. for f:

$$\alpha = \frac{q(f|f^*)}{q(f^*|f)} \frac{P(p)P(f^*P(U|f^*)P(Y|U,p)}{P(p)P(f)P(U|f)P(Y|U,p)}$$

Naive Metropolis-Hastings for f, p, and U

Pretty similar to the 2D single component-wise MH algorithm, with Hastings ratios:

3. for U, with n separate updates for $U_i \sim \text{Bernouilli}(f_{current})$, $i = 1 \dots n$:

$$\alpha = \frac{q(U_i|U_i^*)}{q(U_i^*|U_i)} \frac{P(p)P(f)P(U_i^*|f)P(Y|U_i^*, p^*)}{P(p)P(f)P(U_i|f)P(Y|U_i, p)}$$

Note the cancellations in all these Hastings Ratios

Gibbs sampling

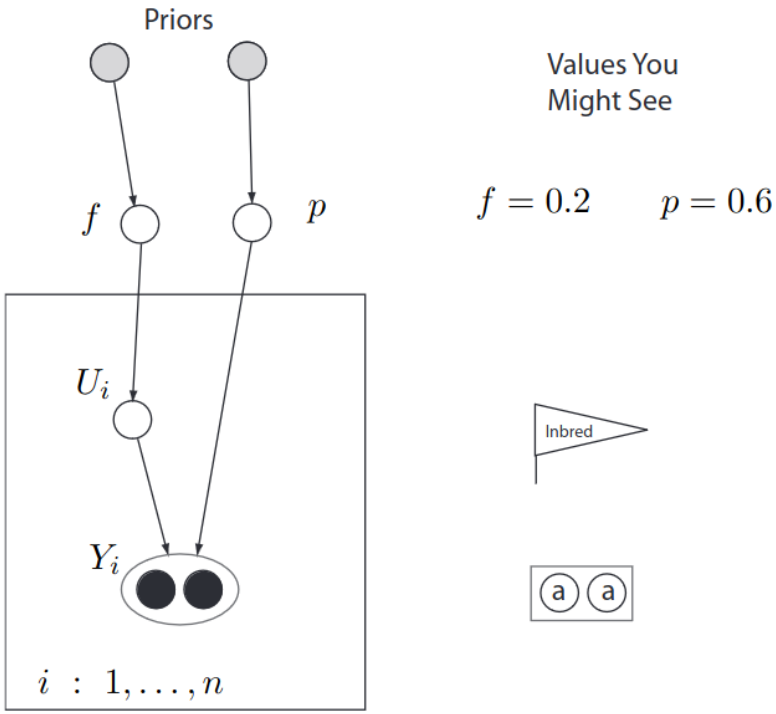
Gibbs sampling is a special form of the component-wise M-H sampler in which **the proposal distribution is the full conditional distribution**

The full conditional distribution for f is the distribution of f conditional upon the current values of all other variables in the model. This must be proportional to the product of all the factors in the joint density that have f in them: $P(f)P(U|f)$:

- The $P(U|f)$ portion of the joint posterior pertains to "counting up our flags"; $P(f)$ is the (Beta) prior for f
- Hence the full conditional for f is a Beta distribution: the "posterior" for f given the "data" U
- And the Hastings ratio is: (and this simplifies A LOT)

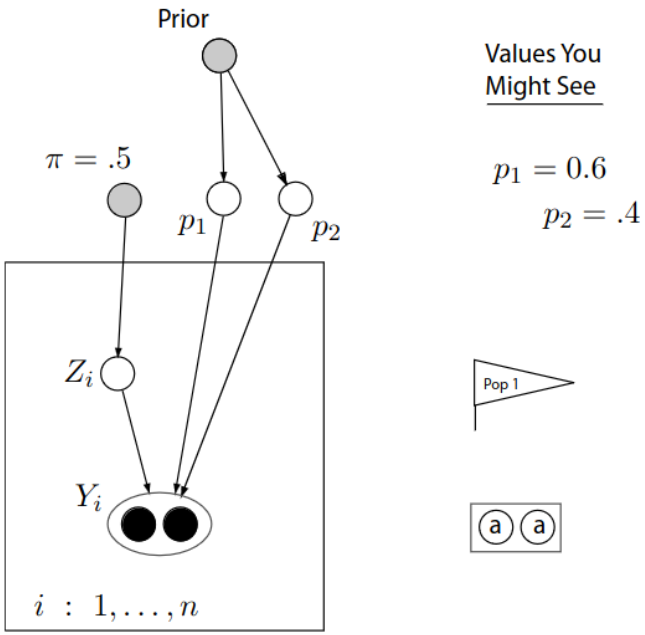
$$\alpha = \frac{P(f)P(U|f)}{P(f^*)P(U|f^*)} \frac{P(p)P(f^*)P(U|f^*)P(Y|U,p)}{P(p)P(f)P(U|f)P(Y|U,p)} = 1(!!!)$$

A Directed Graphical View of The Model



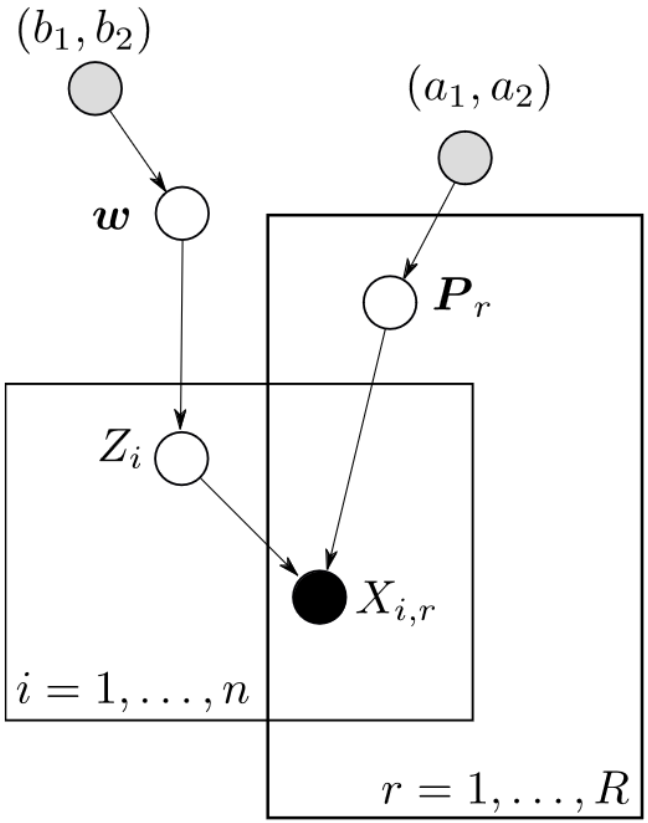
But the inference is conditional upon the underlying model

We could have attributed the departure from Hardy-Weinberg proportions to a mixture (in the proportions of $\pi = 0.5$) of two populations with allele frequencies p_1 and p_2, respectively



This is the *structure* model with no admixture and $K = 2$!

The haploid genetic mixture model as a DAG



MCMC convergence

- MCMC rely on the asymptotic (i.e. after infinitely many iterations) sampling properties of the simulated values from the Markov chain: they should behave as if they were sampled from the joint posterior distribution we are interested in.
- In practice we cannot perform infinitely many iterations before we regard the simulated values as coming from the posterior distribution. Thus we "hope" that provided we do enough iterations our simulated values will be sufficiently close to coming from the posterior distribution. But how do we know when we are sufficiently close?

MCMC convergence

- Some formal methods exist: Gelman-Rubin R statistic; stationarity (plot and others)
- We recommend:
  - Run the MCMC sampler multiple times from different starting points and compare
  - Plot trace plots of different parameters against iteration number: do they look "hairy"?
  - Also trace-rank plots.
  - R package coda
  - Think out of the box in relationship to your posterior target. Multimodal? What about the variance of the proposal?
- A slightly related topic, the MCMC efficiency $\rightarrow$ auto-correlation (and effective sample size), acceptance rates

Thanks a lot for your attention.

It has been a privilege working with you during this workshop!