Lecture 9 - Gibbs Sampling + Rejection Methods

Previously: MH-MCMC Example - Bivariate normals

- Assume we have some data from a bivariate normal distribution, for which we wish to estimate the means.
- For convenience, we will assume the variance/covariance structure is known.

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_2\sigma_1 & \sigma_2^2 \end{pmatrix}\right]$$

Properties:

$$X_1 \sim N(\mu_1, \sigma_1^2)$$

$$X_2 \sim N(\mu_2, \sigma_2^2)$$

$$Correlation(X_1, X_2) = \rho. \text{ [Recall } \rho_{X_1X_2} = \frac{Cov(X_1X_2)}{\sigma_X\sigma_Y}]$$

In-class exercise

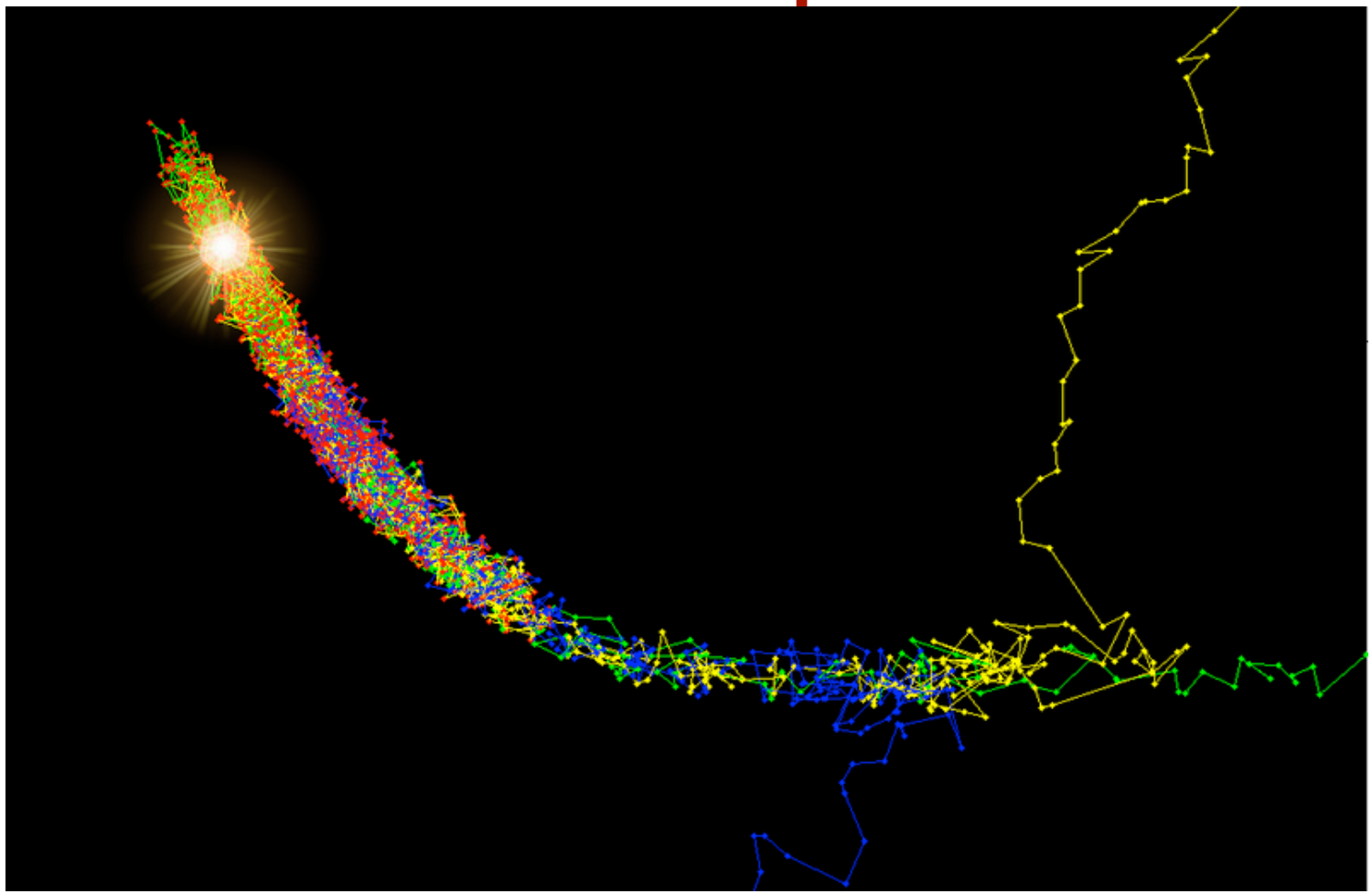
- Try running the algorithm for each of the three test datasets:

```
mu.vector <- c(3, 1)    # the vector of means for the multi-variate normal
variance.matrix <- cbind(c(1, 0), c(0, 4))   # the variance-covariance matrix for the multi-variate normal
# variance.matrix <- cbind(c(1, 1.5), c(1.5, 4))
# variance.matrix <- cbind(c(1, 1.98), c(1.98, 4))
```

- Explore behavior (use Gelman diagnostics: acfs, posterior densities) for each case.
- Fix it, for cases in which it performs badly.

Assignment 3 due next week

MCMC in pictures



The result of three Markov chains running on the 3D Rosenbrock function using the Metropolis-Hastings algorithm. The algorithm samples from regions where the posterior probability is high and the chains begin to mix in these regions. The approximate position of the maximum has been illuminated. Note that the red points are the ones that remain after the burn-in process. The earlier ones have been discarded.
(wikipedia)

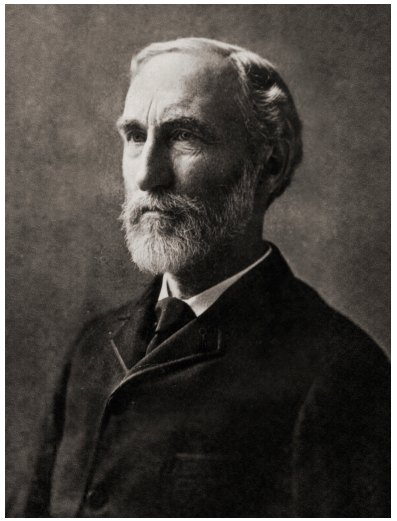
MCMC - Gibbs sampling

- Suppose $\Theta=(\theta_1,\theta_2,...,\theta_k)$
- Suppose that we can write $f(\theta_i|\theta_1,...,\theta_{i-1},\theta_{i+1},...,\theta_k)$ for all i.
- Then, at each iteration, use following algorithm:
  1. Propose $\theta_1' \sim f(\theta_1|\theta_2,...,\theta_k)$
  2. Propose $\theta_2' \sim f(\theta_2|\theta_1',\theta_3,...,\theta_k)$
  3. ...
  n. Propose $\theta_k' \sim f(\theta_k|\theta_1',...,\theta_{k-1}')$

Result: samples from joint distribution, $f(\theta_1,\theta_2,\ldots,\theta_k)$

- WinBUGS software (http://www.mrc-bsu.cam.ac.uk/bugs/)
- OpenBUGS software (http://www.openbugs.net/w/FrontPage)
- JAGS software (http://mcmc-jags.sourceforge.net/)

What happened to the acceptance prob. (i.e. Hastings Ratio)?



Josiah Willard Gibbs	
Born	February 11, 1839 New Haven, Connecticut,
Died	April 28, 1903 (aged 64) New Haven, Connecticut,
Fields	Physics, chemistry, mathematics

Gibbs sampling

$$f(A \mid B) = \frac{f(A \cap B)}{f(B)}$$

$$
\begin{aligned}
h &= \min\left\{1, \frac{f(\Theta')q(\Theta' \to \Theta)}{f(\Theta)q(\Theta \to \Theta')}\right\} \\
&= \min\left\{1, \frac{f(\Theta')f(\theta_i \mid \theta'_1, \theta'_2, ..., \theta'_{[-i]}, ..., \theta'_k)}{f(\Theta)f(\theta'_i \mid \theta_1, \theta_2, ..., \theta_{[-i]}, ..., \theta_k)}\right\} \\
&= \min\left\{1, \frac{f(\Theta')f(\theta'_1, \theta'_2, ..., \theta_i, ..., \theta'_k)/f(\theta'_1, \theta'_2, ..., \theta'_{[-i]}, ..., \theta'_k)}{f(\Theta)f(\theta_1, \theta_2, ..., \theta'_i, ..., \theta_k)/f(\theta_1, \theta_2, ..., \theta_{[-i]}, ..., \theta_k)}\right\} \\
&= \min\left\{1, \frac{f(\Theta')f(\Theta)/f(\theta'_1, \theta'_2, ..., \theta'_{[-i]}, ..., \theta'_k)}{f(\Theta)f(\Theta')/f(\theta_1, \theta_2, ..., \theta_{[-i]}, ..., \theta_k)}\right\} \\
&= 1 \qquad \text{since } \Theta' \text{ and } \Theta \text{ differ only at } \theta_i \neq \theta'_i.
\end{aligned}
$$

So the Hastings Ratio equals 1 (by construction)

MCMC Application (Gibbs sampling):STRUCTURE

- Problem: To 'untangle' population structure in genetic data
- Goal: To assign each individual's genotype to populations
- Method: Model-based, uses 'unlinked' (i.e., independent) loci.
- MCMC example: "Inference of Population Structure Using Multi-locus Genotype Data" - Jonathan K. Pritchard, Matthew Stephens and Peter Donnelly, Genetics 155: 945–959 (2000).



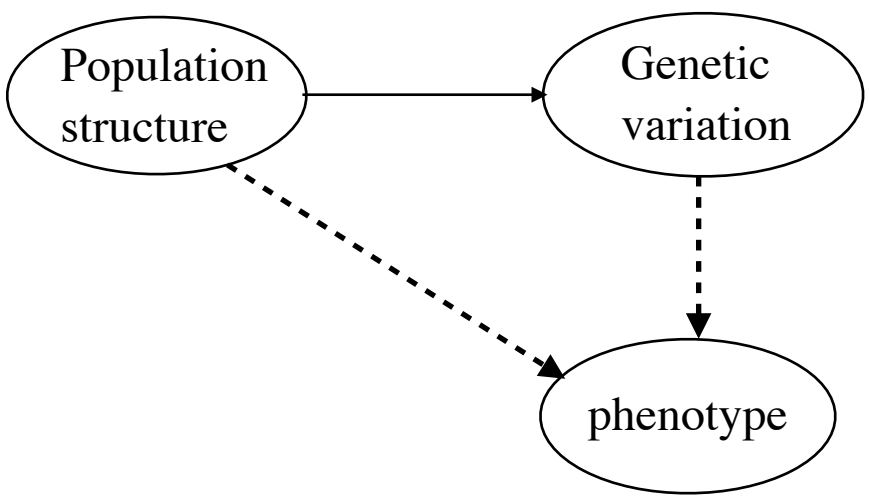
Why care? - Genome-wide association studies

0 1 1 1 1 1 0 1 0 2 1 2 2 0 1 0 0 0 1 1	Control
2 0 1 1 1 2 0 0 0 1 0 1 1 0 1 1 0 1 0 0	Control
2 0 1 2 2 0 1 2 1 0 0 1 1 0 1 0 0 1 1 1	Control
1 2 1 1 2 1 1 1 1 0 1 1 1 0 0 2 2 2 0 2	Control
1 1 2 1 0 1 2 1 1 1 1 2 1 2 1 2 1 2 1 1	Case
2 2 1 2 0 1 0 0 0 1 2 2 1 2 1 2 1 0 2 1	Case
0 1 1 0 0 2 1 0 0 2 1 1 1 2 1 1 2 0 1 0	Case
0 1 1 0 0 1 0 2 2 1 1 1 1 2 0 1 2 1 1 2	Case

"0" = no copies of mutant allele at that position
"1" = one copy of mutant allele at that position
"2" = two copies of mutant allele at that position

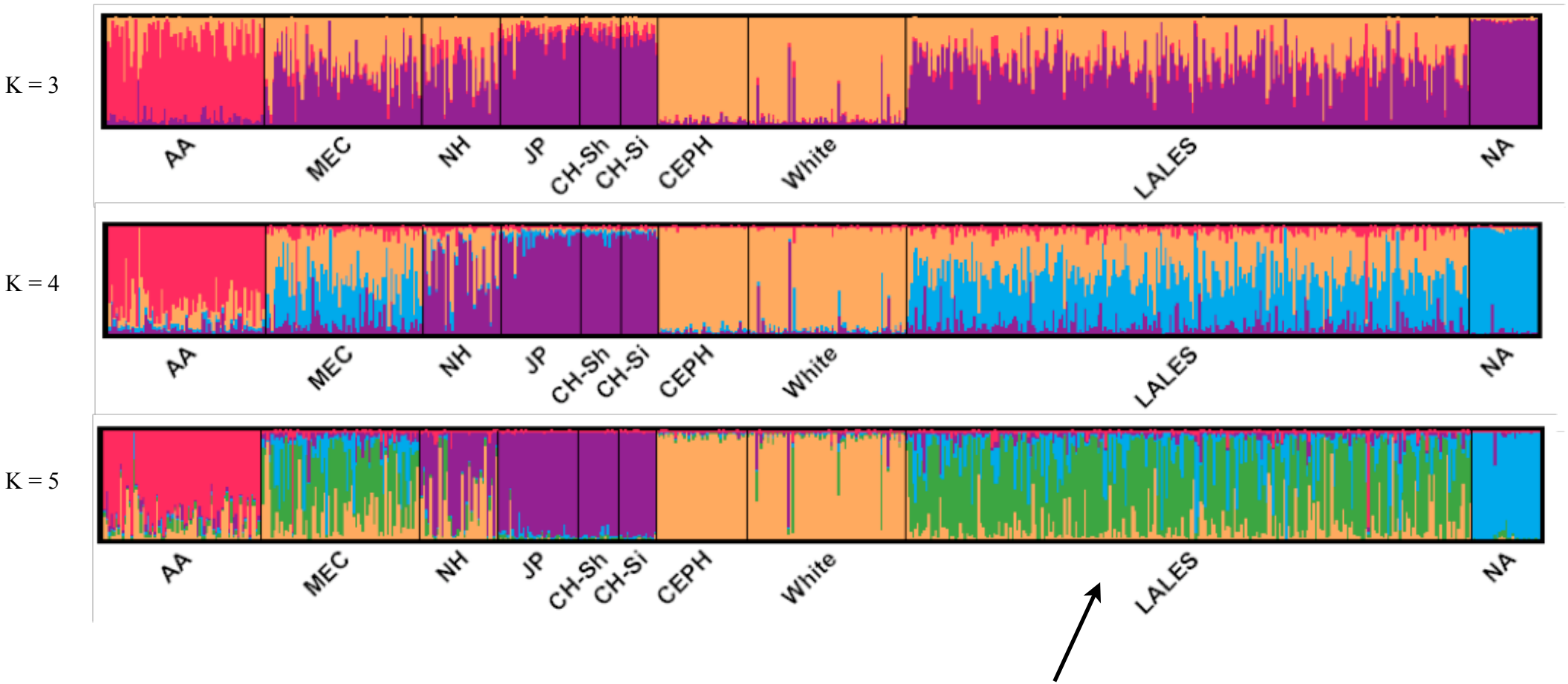
Why care?

- Population structure can cause inflation of false-positive rate in GWAS.
- Common to incorporate effects of stratification in an analysis.
- Confounding triangle:



Zhao K, Aranzana MJ, Kim S, et al. An Arabidopsis example of association mapping in structured samples. *PLoS Genet*. 2007;3(1):e4.

Understanding structure:
Los Angeles Latino Eye Study



Single population?

Population 1

$p_{111}, p_{121}, p_{131}, p_{141}$
$p_{112}, p_{122}, p_{132}, p_{142}$

Population 2

$p_{211}, p_{221}, p_{231}, p_{241}$
$p_{212}, p_{222}, p_{232}, p_{242}$

Population 3

$p_{311}, p_{321}, p_{331}, p_{341}$
$p_{312}, p_{322}, p_{332}, p_{342}$

p_{klj} = frequency of allele j at locus l in population k, where k=1, 2, . . . , K and j =1, 2, . . . , J.
(NB. $p_{kl2} = 1\text{-}p_{kl1}$, for a 2-allele model, such as Single Nucleotide Polymorphism [SNP] data.)

Notation:

X: genotypes of sampled indivs.

Z: unknown population of origin

P: allele frequencies in source populations.

Model assumptions:

- Hardy-Weinberg equilibrium within each population [so, P(ab)=2P(a)P(b)]
- Linkage equilibrium between markers within populations.
- Thus, each allele at each locus in each genotype is an independent draw from the appropriate frequency distribution [Pr(X |Z, P) is completely specified].

X: genotypes of sampled indivs.

Z: unknown population of origin

P: allele frequencies in source populations.

- Specify priors: Pr(Z), Pr(P)
- Observe: genotypes X
- Knowledge about Z and P given by

$$\Pr(Z, P \mid X) \propto \Pr(Z)\Pr(P)\Pr(X \mid Z, P) \qquad (1)$$

Calculated via Gibbs sampling MCMC!

Case 1: no admixture

- **Each individual assumed to be drawn from a single source population.**
- Sample N indivs. at L loci

  $(x_j^{(i,1)}, x_j^{(i,2)})$ = genotype of i^{th} indiv. at j^{th} locus

  $z^{(i)}$= population from which indiv i originated

  p_{klj} = frequency of allele j at locus l in population k, (k=1, 2, . . . , K and j =1, 2, . . . , J_l; J_l is the number of distinct alleles observed at locus l).

X: genotypes of sampled indivs.

Z: unknown population of origin

P: allele frequencies in source populations.

- Conditional on pop. of origin of an indiv., genotypes are assumed drawn independently at random from the corresponding pop. frequency distribution:

$$\Pr(x_j^{(i,a)} = m \mid Z,P) = p_{z^{(i)}lm} \qquad (2)$$

[$z^{(i)}$ = pop. of origin of indiv. i]

- Prior for origin: $\Pr(z^{(i)}=k) = 1/K$ (3) (easy to extend)
- Prior for allele frequencies: Dirichlet distribution $\mathcal{D}(\lambda_1,\lambda_2,\ldots,\lambda_J)$ (generalization of Beta distribution). $\lambda_i \propto$ mean value for freq. i. Setting $\lambda_1=1$ for all i gives joint uniform dist. on allele freqs.

X: genotypes of sampled indivs.

Z: unknown population of origin

P: allele frequencies in source populations.

MCMC - Algorithm 1

Starting with initial values $Z^{(0)}$ for Z (by drawing $Z^{(0)}$ at random using (3) for example), **iterate** the following steps for m= 1,2, . . . :

1. Sample $P^{(m)}$ from $Pr(P \mid X, Z^{(m-1)})$.
2. Sample $Z^{(m)}$ from $Pr(Z \mid X, P^{(m)})$

Result: samples from Pr(Z,P | X)
[Gibbs sampling]

X: genotypes of sampled indivs.

Z: unknown population of origin

P: allele frequencies in source populations.

Details - step 1

- Sample $p_{kl.}$ independently, for each (k,l) as
$(p_{kl.}|X,Z) \sim D(\lambda_1+n_{kl1},\lambda_2+n_{kl2},\ldots,\lambda_{J_l}+n_{kl1})$

where $n_{klj} = \#\{(i,a) : x_l^{(i,a)}=j \text{ and } z^{(i)}=k\}$ [the number of copies of allele j at locus l observed in individuals assigned (by Z) to population k]

Notation reminder:

p_{klj} = frequency of allele j at locus l in population k

$(x_j^{(i,1)},x_j^{(i,2)})$ = genotype of i^{th} indiv. at j^{th} locus

$z^{(i)}$= population to which indiv. i assigned

X: genotypes of sampled indivs.

Z: unknown population of origin

P: allele frequencies in source populations.

Details - step 2

- Simulate $z^{(i)}$ independently, for each i, from:

$$\Pr(z^{(i)} = k \mid X, P) = \frac{\Pr(x^{(i)} \mid P, z^{(i)} = k)}{\sum_{k'=1}^{K} \Pr(x^{(i)} \mid P, z^{(i)} = k')}$$

where

$$\Pr(x^{(i)} \mid P, z^{(i)} = k) = \prod_{l=1}^{L} p_{klx^{(i,1)}} p_{klx^{(i,2)}}.$$

This is where they use that the loci are unlinked

Reminder:

p_{klj} = frequency of allele j at locus l in population k

$(x_j^{(i,1)}, x_j^{(i,2)})$ = genotype of i^{th} indiv. at j^{th} locus

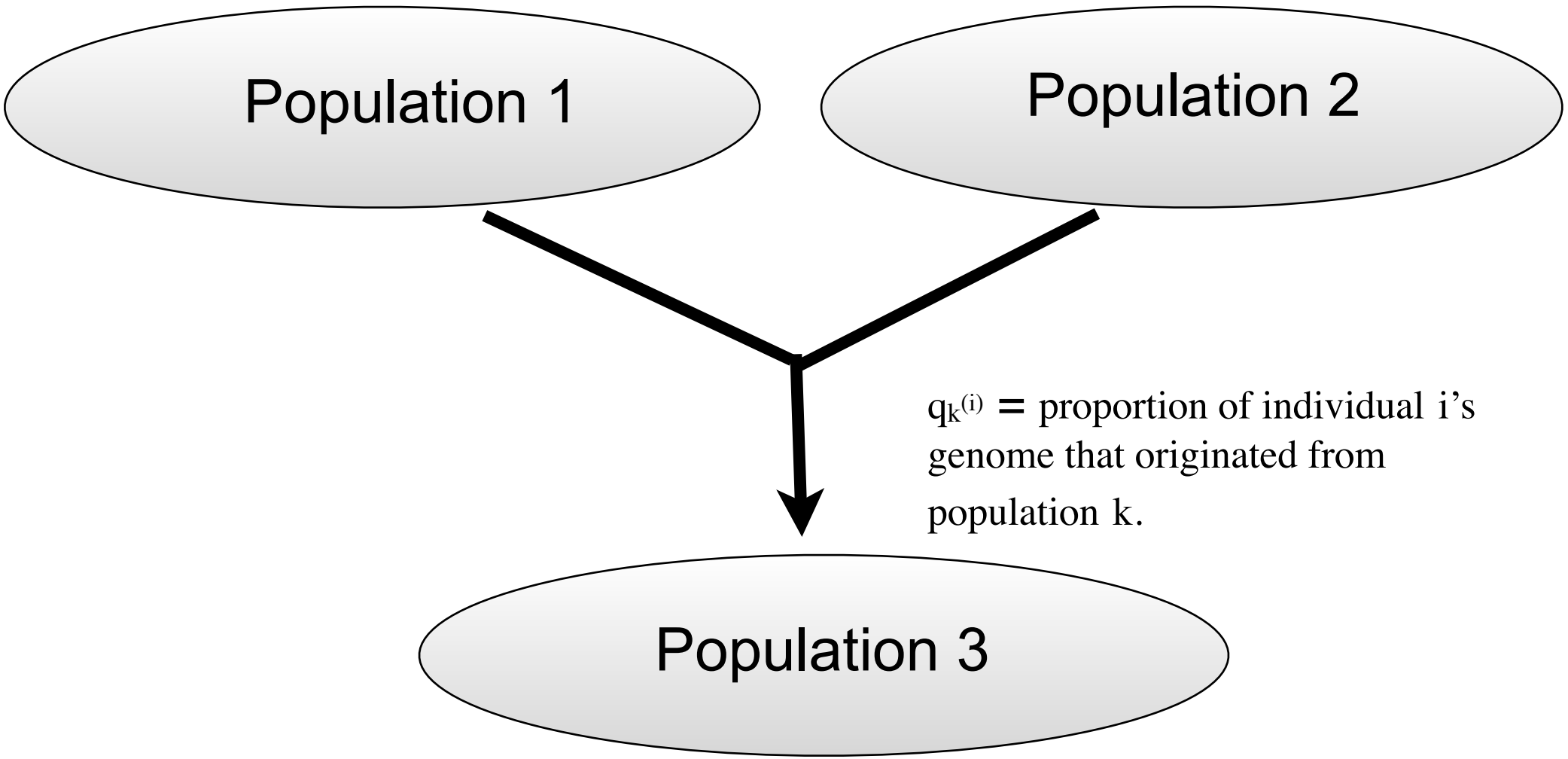
$z^{(i)}$ = population to which indiv. i assigned

X: genotypes of sampled indivs.

Z: unknown population of origin

P: allele frequencies in source populations.

Admixture


Population 1
Population 2
Population 3
$q_k^{(i)}$ = proportion of individual i's genome that originated from population k.

- No longer true that entire indiv. comes from a given source pop. So...

  $z_l^{(i,a)}$ = population of origin of allele copy $x_l^{(i,a)}$.

- Goal is now to estimate Q (ancestry proportions of each individual).

- Use same probability model as before, where everything is now locus-specific.

- Need a model for Q: Dirichlet distribution.

  $q^{(i)} \sim \mathcal{D}(\alpha,\alpha,...,\alpha)$ [$\alpha>>1$, indivs. have material from each of the K pops; $\alpha\sim0$, no admixture.]

Gibbs sampling:

Step 1. Sample $P^{(m)}$, $Q^{(m)}$ from $\Pr(P, Q \mid X, Z^{(m=1)})$.
[estimate allele frequencies for each population and admixture proportions of each individual, assuming origin of each allele is known]

Step 2. Sample $Z^{(m)}$ from $\Pr(Z \mid X, P^{(m)}, Q^{(m)})$.
[estimate the population of origin of each allele copy, assuming that the population allele frequencies and the admixture proportions are known]

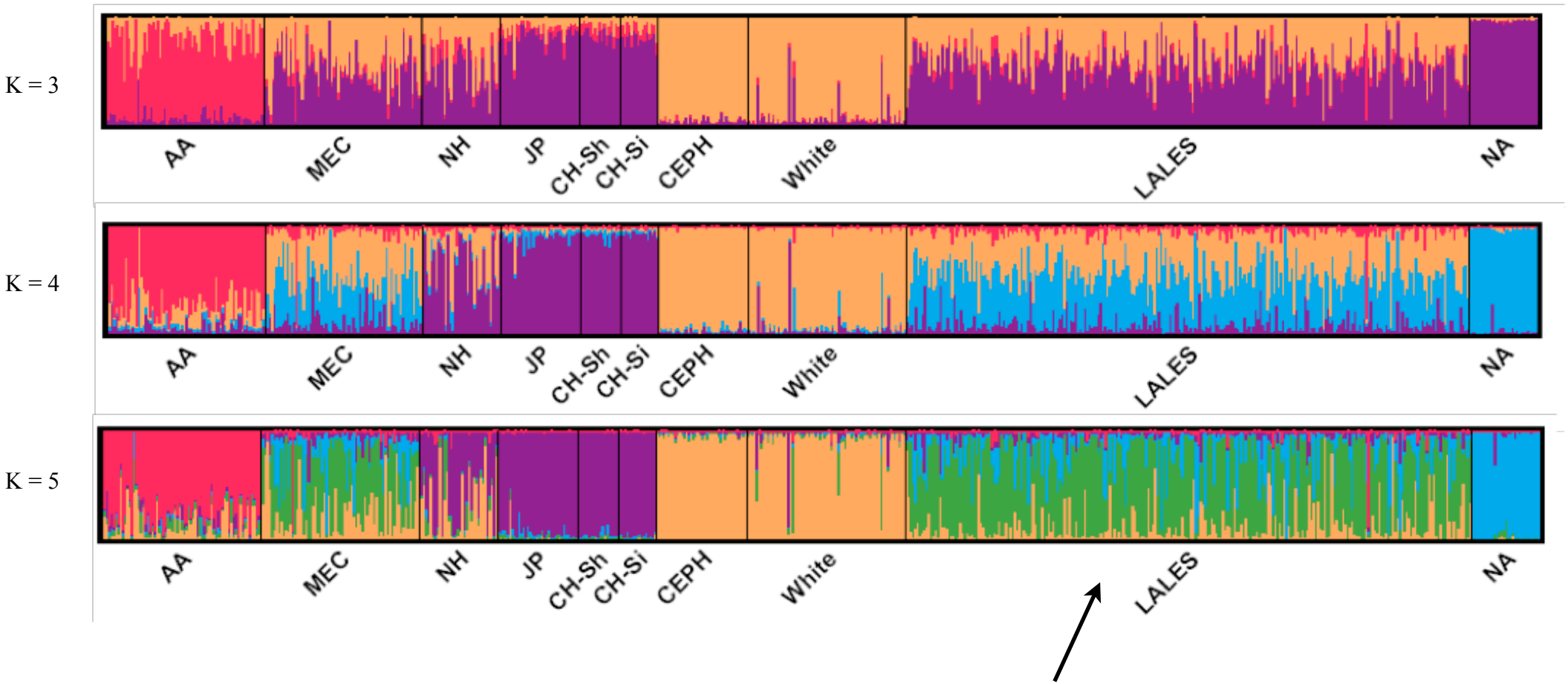
X: genotypes of sampled indivs.

Z: unknown population of origin

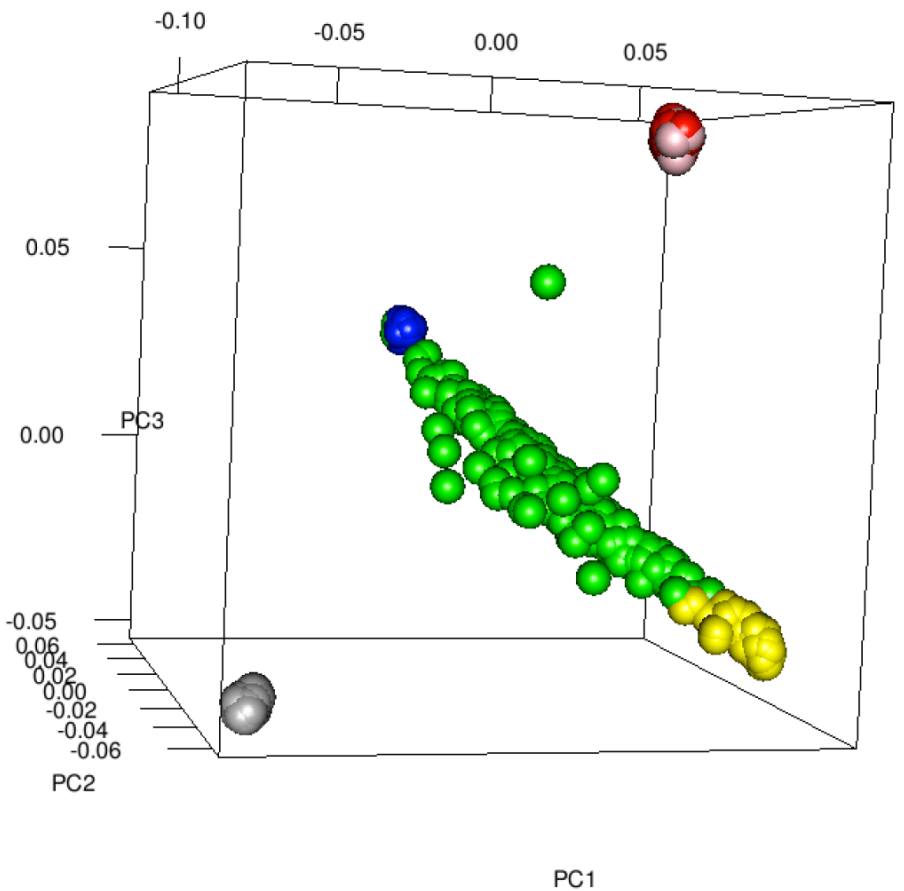
P: allele frequencies in source populations.

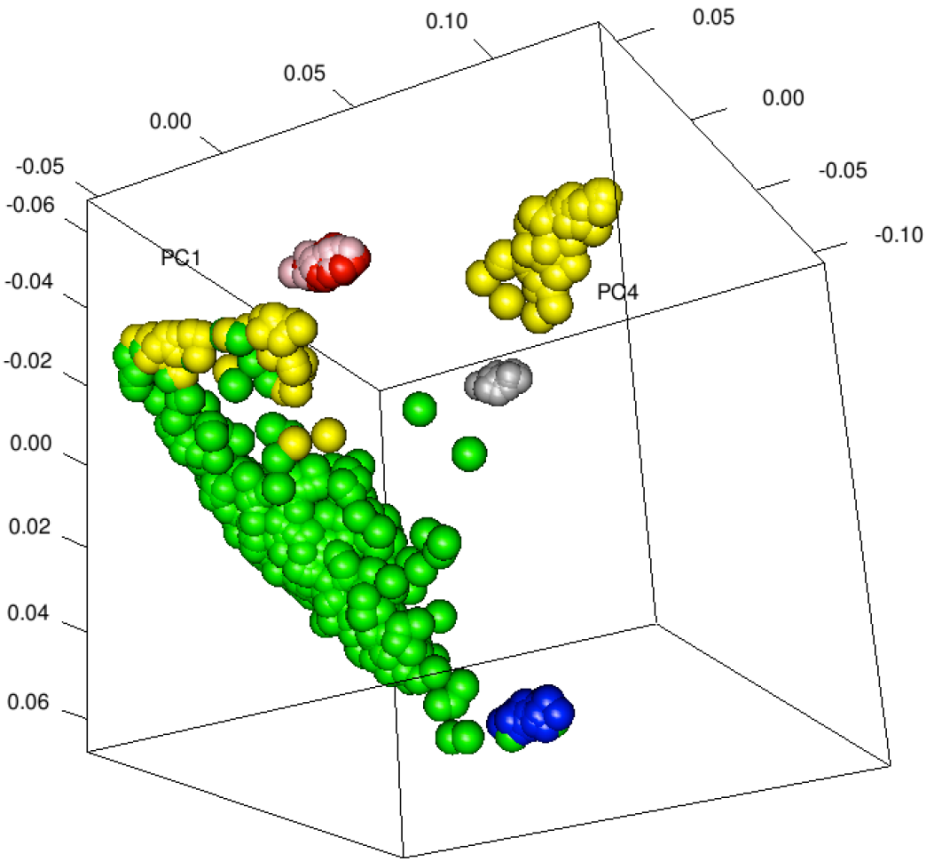
Q: admixture proportions

Understanding structure:
Los Angeles Latino Eye Study



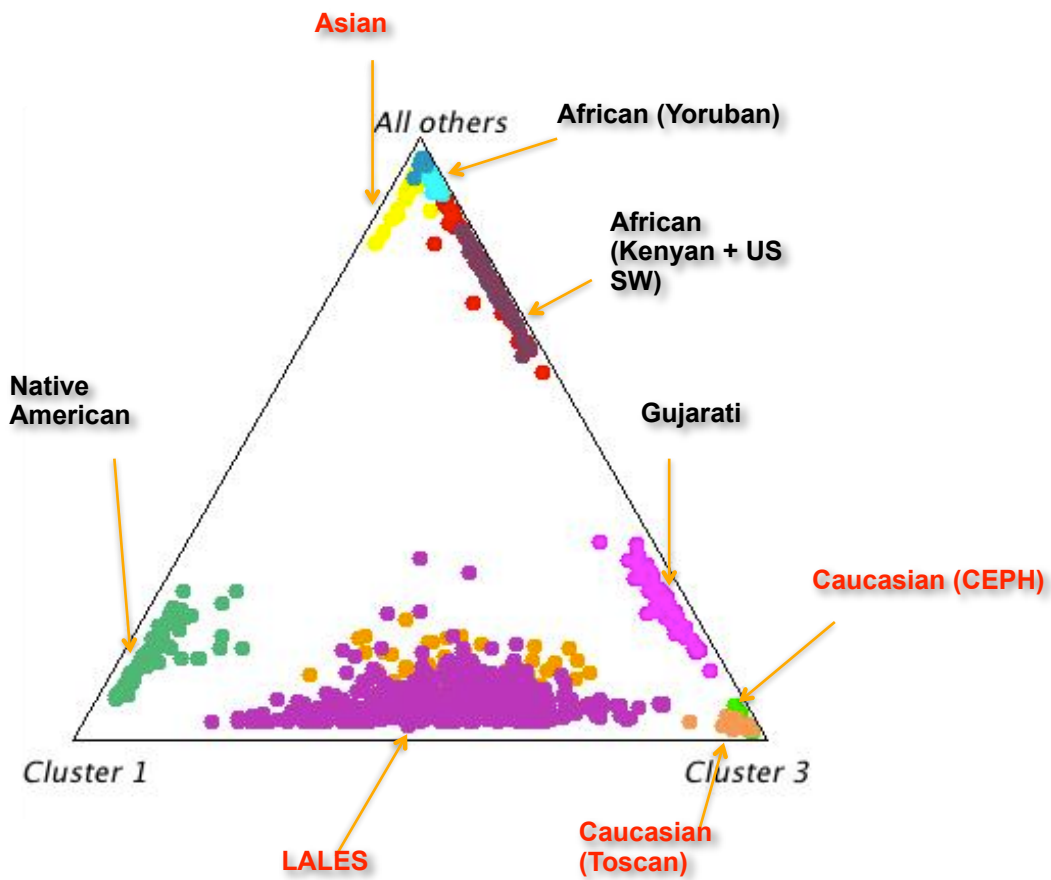
PCA plots


-0.10
-0.05
0.00
0.05
0.05
0.00
PC3
-0.05
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
PC2
PC1


0.10
0.05
0.00
-0.05
-0.06
-0.04
-0.02
0.00
0.02
0.04
0.06
PC1
0.05
0.00
-0.05
-0.10
PC4

Ancestry estimates

- Overall ancestry LALES:
  - 44.5% Native Amer.
  - 50.8% Caucasian
  - 4.5% African
  - 0.3% Asian



Gibbs sampling - simple example

$$(\theta, \mu) \sim N\left(\begin{matrix}0\\0\end{matrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

- Given θ_t, μ_t:

1. Generate $\theta_{t+1} \sim N(\rho\mu_t, 1-\rho^2)$
2. Generate $\mu_{t+1} \sim N(\rho\theta_{t+1}, 1-\rho^2)$

- Result: samples from joint distribution (θ, μ).

Non-examinable lab exercise

- Code up the Gibbs sampler to simulate samples from:

$$(\theta, \mu) \sim N\left(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

- Compare this to performance of a standard Metropolis-Hastings MCMC algorithm, particularly for high values of ρ. [look at autocorrelation between θ values (or μ values)].
- R library: mvtnorm [dmvnorm(x, mean = rep(0, p), sigma = diag(p), log = FALSE)]
- If you have time, also implement an adaptive MCMC scheme to see how it compares with the Gibbs sampler in terms of efficiency.

RJAGS

- RJAGS is an R implementation of JAGS (Just Another GIBBS sampler")
- See example on GitHub (Week9-RJAGS-Example)

Rejection methods

Reminder: How to Sample From a Random Variable

- Want to sample from a density f(x).
- Suppose we can calculate the cumulative density function F(x).

1. Generate u~Unif[0,1]
2. Set $u=F(x)$
3. Solve for x
4. x has density f.

The "inversion" method

e.g. exponential distribution. $F(x)=1-e^{\lambda x}$, so $u= 1-e^{\lambda x}$,
so $x= \log(1-u)/\lambda$.

What if we cannot calculate the CDF F? (e.g. urns, trees,...)

Previously: Monte Carlo simulation methods, MCMC

This week: Rejection methods (accept/reject algorithms) + Importance sampling.

Rejection method

Goal: Generate samples from f(), defined over a domain [a,b].

Suppose: We know $f(x) \leq K$ for all $x \in [a,b]$.

Use the following iterative scheme:

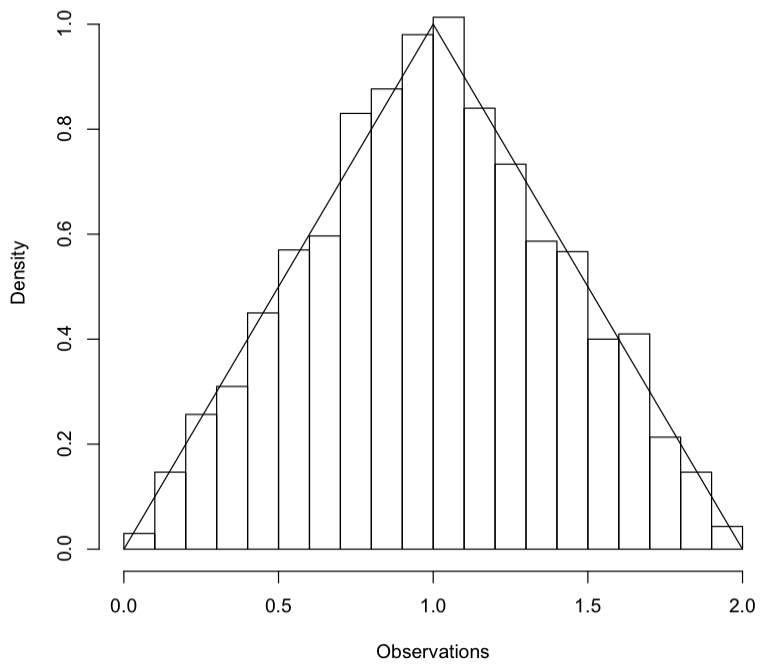
1. Sample $x \sim \text{Unif}[a,b]$
2. Accept x with probability $f(x)/K$.
3. Return to 1.

Results: independent samples from f().

```
# program spuRs/resources/scripts/rejecttriangle.r
rejectionK <- function(fx, a, b, K) {
  # simulates from the pdf fx using the rejection algorithm
  # assumes fx is 0 outside [a, b] and bounded by K
  # note that we exit the infinite loop using the return statement
  while (TRUE) {
    x <- runif(1, a, b)
    y <- runif(1, 0, K)
    if (y < fx(x)) return(x)
  }
}
fx<-function(x){
  # triangular density
  if ((0<x) && (x<1)) {
    return(x)
  } else if ((1<x) && (x<2)) {
    return(2-x)
  } else {
    return(0)
  }
}
# generate a sample
set.seed(21)
nreps <- 3000
Observations <- rep(0, nreps)
for(i in 1:nreps)   {
  Observations[i] <- rejectionK(fx, 0, 2, 1)
}
# plot a scaled histogram of the sample and the density on top
hist(Observations, breaks = seq(0, 2, by=0.1), freq = FALSE,
     ylim=c(0, 1.05), main="")
lines(c(0, 1, 2), c(0, 1, 0))
```

rejecttriangle.r on Github
(In Week9-Rejection-Methods)

Example: Sampling from a triangular distribution

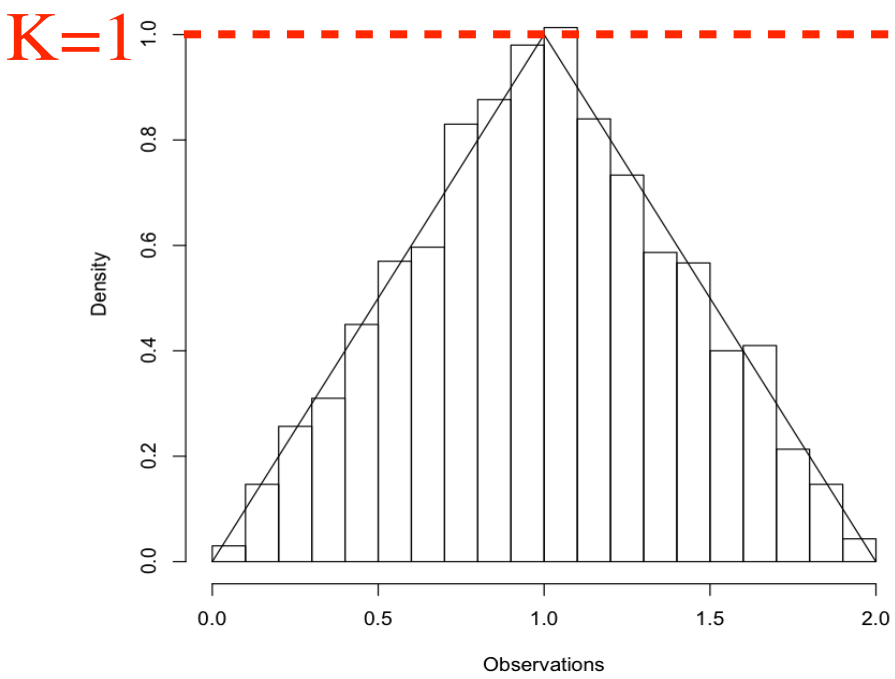


Overview

- We can view this as simulating data points (x,y) uniformly from the rectangle that covers [a,b] on the x-axis and [0,K] on the y-axis, and then rejects the x-value if y>f(x)

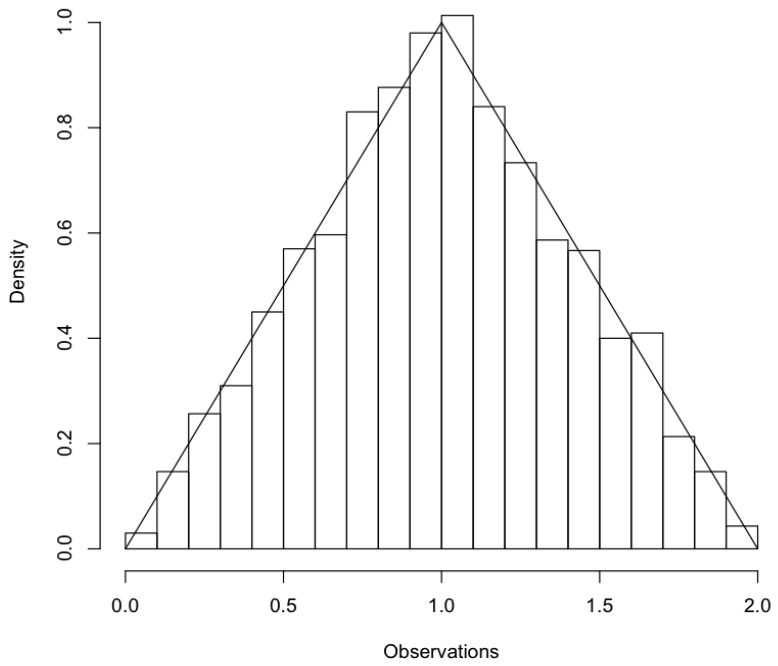
- What happens if a,b or K equals +/- infinity?

$$U < f(x)/K \iff KU < f(x)$$



For triangle distribution:
We used K ≥ 1 (i.e. a rectangle distribution)

K=1.5 ----------



For triangle distribution:
We used K ≥ 1 (i.e. a rectangle distribution)

General Rejection method

Goal: Generate samples from f(), defined over a domain [a,b].

Suppose:

a. We have an envelope density h() from which we can simulate.

b. There exists an $K<\infty$ such that $\sup_x f(x)/h(x) \leq K$

Use the following iterative scheme:

So $f(x) \leq Kh(x)$

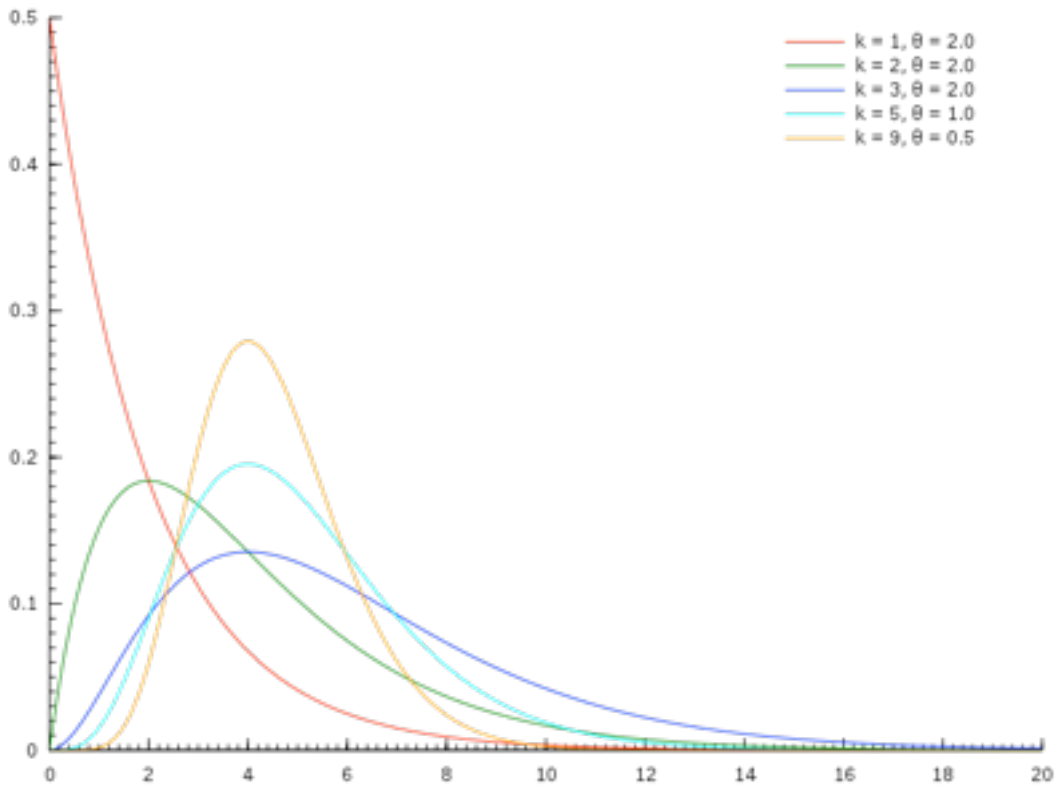
1. Simulate *x* from h().
2. Generate y from Unif[0,Kh(x)]. If y<f(x) then accept x.
3. Return to 1.

Results: independent samples from f().

Previously, our envelope was a rectangle.

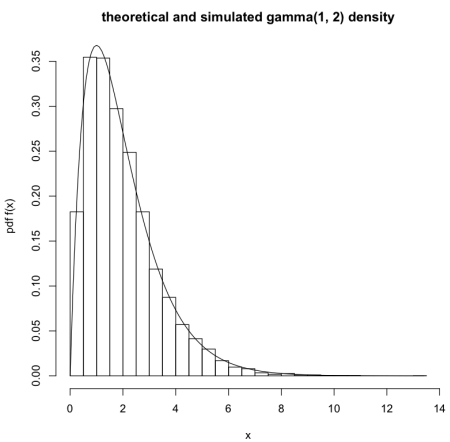
Example: Gamma distribution Γ(λ,m)

- Gamma density: $f(x)=\lambda^m x^{m-1} e^{-\lambda x}/\Gamma(m)$, for $x>0$.
- There is no formula for the cumulative density function.
-



Non -examinable exercise

- Write a rejection algorithm that uses the Uniform (rectangle) envelope, with K=1, to generate 10000 samples from a $\Gamma(\lambda,m) = \Gamma(1,2)$ distribution.
  - Plot your results.
  - How many iterations were needed to produce 1000 samples?



- Write a rejection algorithm that uses the exponential envelope $h(x)=\mu e^{-\mu x}$, where $\mu=\lambda/m$ to generate 10000 samples from a $\Gamma(1,2)$ distribution.
  - Plot your results.
  - How many iterations were needed to produce the samples?
- NB. R's built-in gamma density dgamma uses the parameters the other way around. [So you would plot dgamma(.,2,1)]

What you need:

- Need to find $K^* = \sup_{x>0} f(x)/h(x)$ (where f() is the Gamma distribution and h() is the exponential envelope with rate λ/m).
  - Answer $K^* = m^m e^{-(m-1)}/\Gamma(m)$
  - [So here, $K^* = 2^2 e^{-1}/\Gamma(2) = 4/e$]

Note: $\Gamma(m) = (m-1)!$

- Will also need to sample from h(). Can do that via the inversion method:

Or just use R's rexp() function if you prefer!

  - CDF of h is $1-e^{-\mu x}$
  - So generate u~Unif(0,1) and set $x = -\ln(1-u)/\mu = -\ln(1-u)m/\lambda$
  - (or, equivalently, set $x = -\ln(u)/\mu = -\ln(u)m/\lambda$)

Pseudocode

```
gamma.sim <- function(lambda, m) {
  # sim a gamma(lambda, m) rv using rejection with an exp envelope
  # assumes m > 1 and lambda > 0
# generate f(x)=lambda^m*x^(m-1)*exp(-lambda*x)/gamma(m)     --- the gamma density at x
# generate h(x)=lambda/m*exp(-lambda/m*x)                    ---  the exponential density at x
# generate  k=m^m*exp(1-m)/gamma(m)                  --- in general. for m=2, λ=1 we have k=4/e
  while (TRUE) {              # keep sampling x's from h(x) and testing them until you accept one
    X <- -log(runif(1))*m/lambda            # generate an x from h, the exponential density
    # Generate y from Unif[0,Kh(x)]
    if (Y < f(X)) return(X)
  }
}

set.seed(1999)
n <- 10000        # number of replicates
g <- rep(0, n)      # create somewhere to keep the answers
for (i in 1:n) g[i] <- gamma.sim(1, 2)    # generate your 10000 gamma r.v.s

hist(g, breaks=20, freq=F, xlab="x", ylab="pdf f(x)",
  main="theoretical and simulated gamma(1, 2) density")
x <- seq(0, max(g), .1)
lines(x, dgamma(x, 2, 1))
```

You need to add the code for the uniform version!

PseudoCodeGamma.R on Github week 9 repo

Rejection method - Bayesian restatement

Suppose we have observed data D, and a model with parameter(s) θ that describes how the data got there. Do the following:

1. Generate parameter(s) θ from prior π.
2. Accept θ with probability P(D|θ)
3. Return to 1.

Results: independent samples from the posterior dist^{n}, P(θ|D).
If an upper bound, K, for P(D|θ) is known, replace 2. with
2.' Accept θ with probability P(D|θ)/K

Often, P(D | θ) cannot be computed, so….

Rejection method - Bayesian restatement II

Suppose we have observed data D, and a model with parameter(s) θ that describes how the data got there. Do the following:

1. Generate parameter(s) θ from prior π.
2. Simulate D' using θ.
3. Accept θ if (D'=D)
4. Return to 1.

Result: independent samples from $P(\theta|D)$

[Likelihood simulation - Diggle and Gratton, JRSS B, 46:193-227, 1984.]

No calculation required at all!!

Application - rejection algorithm

http://www.cmb.usc.edu/people/stavare/Tavare.html

https://sites.google.com/site/baldingstatisticalgenetics/

http://www.stats.ox.ac.uk/~griff/

- "Inferring Coalescence Times From DNA Sequence Data" Simon Tavaré, David J. Balding, R. C. Griffiths and Peter Donnelly, Genetics 145 505-518, 1997.
- Goal: estimate time to most recent common ancestor [TMRCA] from Y-chromosome data.
  - Data:
    - Hammer (1995)
    - Whitfield (1995)
- Method: rejection algorithm (accept-reject algorithm) using coalescent model with infinite-sites mutation model

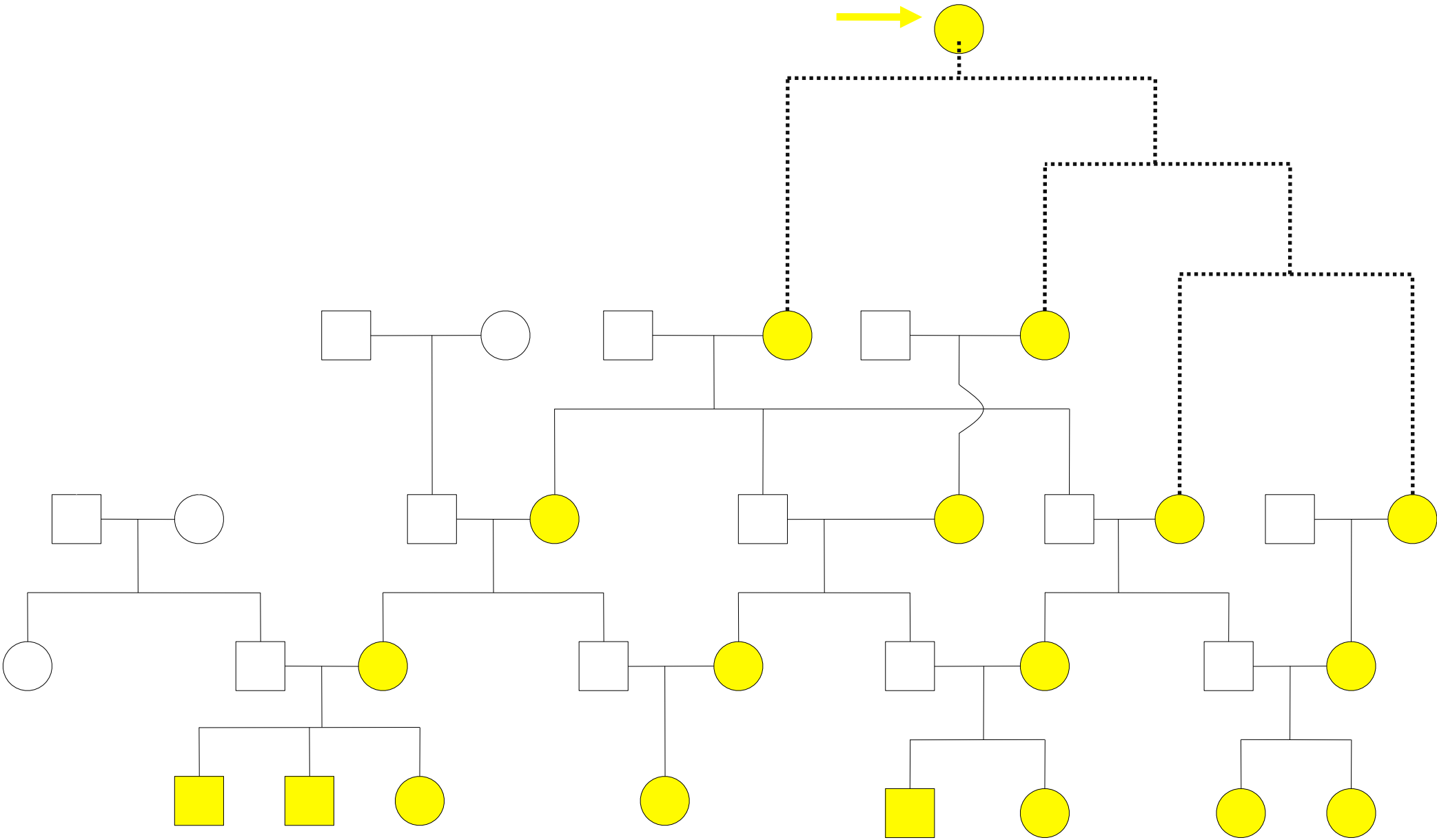
https://www.stats.ox.ac.uk/people/academic_staff/peter_donnelly

Motivation: Time to Most Recent Common Ancestor [MRCA]

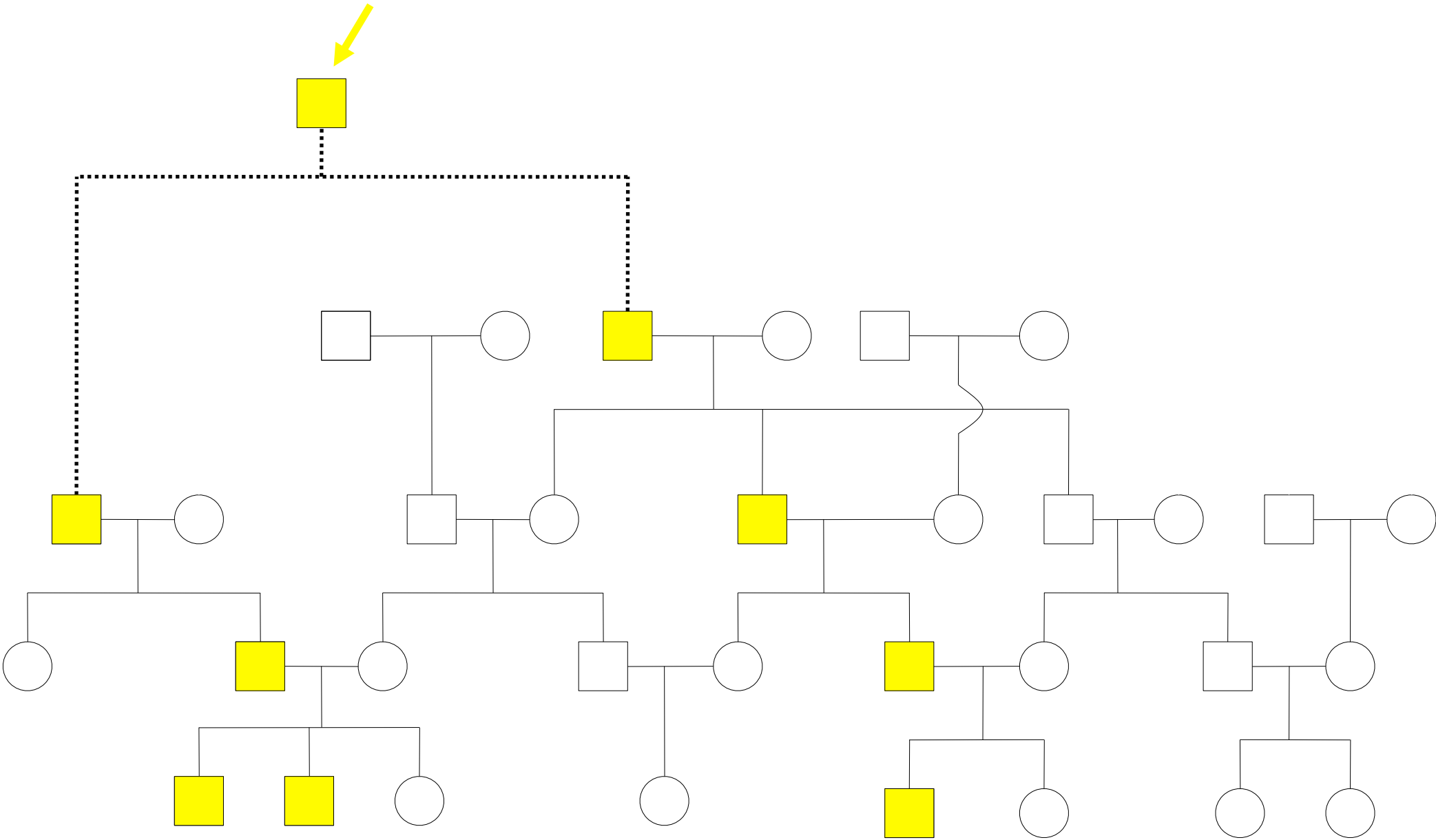
- How long ago was the most recent common ancestor of the human population?

- What do we mean by MRCA?
  - An individual from whom we all inherit everything?
  - An individual from whom we all inherit something?

mt Eve

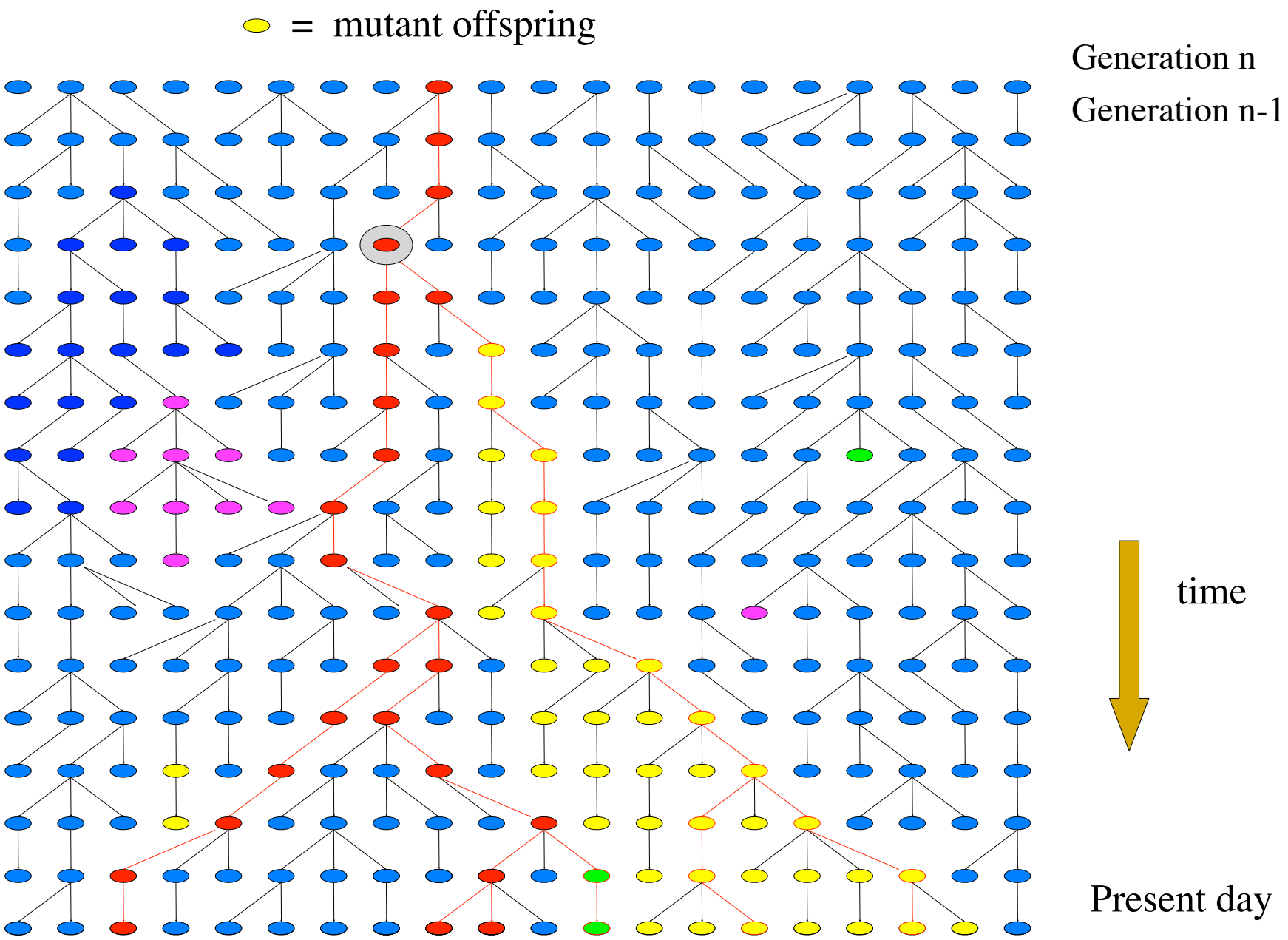


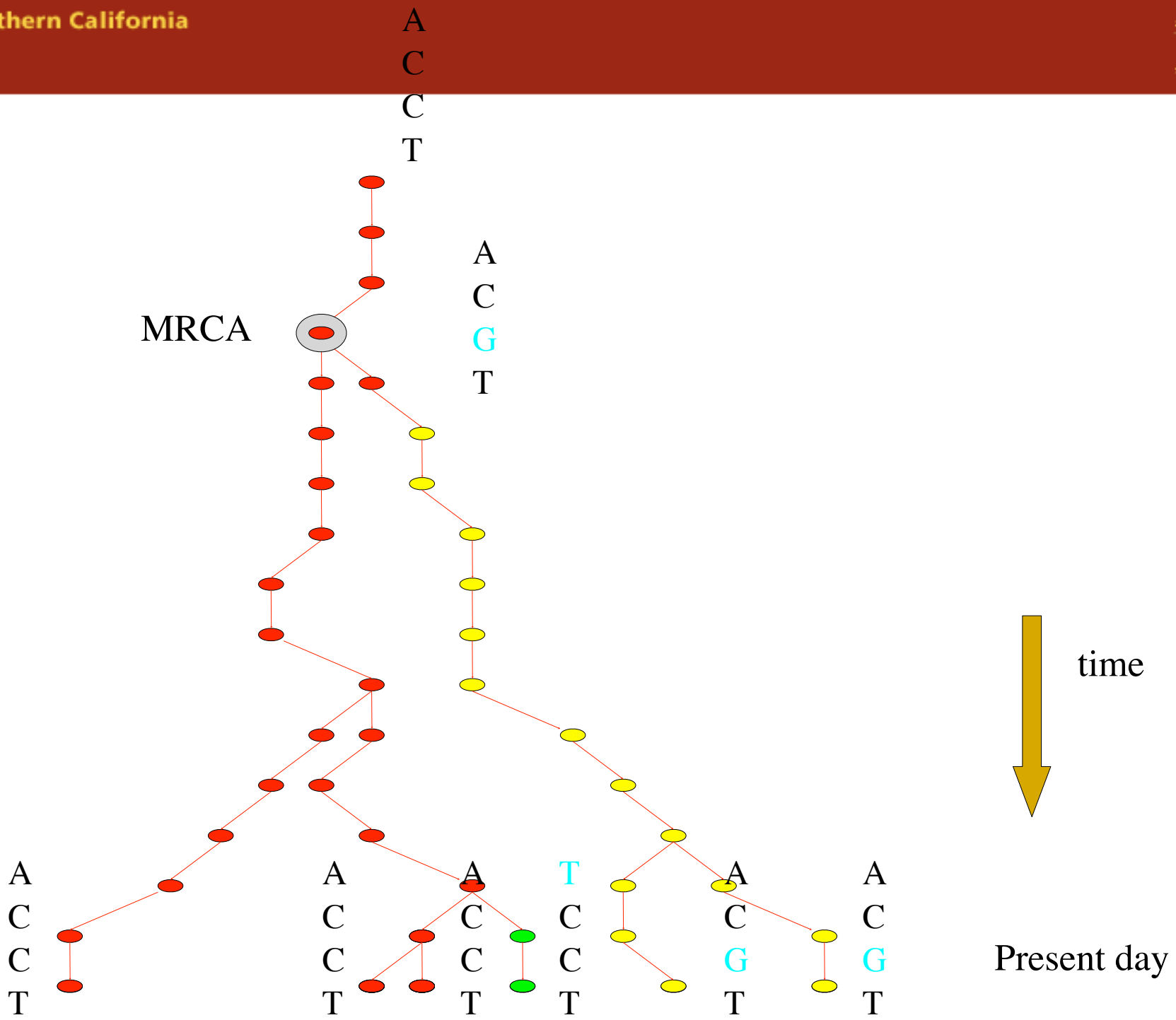
Y-chromosome Adam



‘Current’ beliefs: common ancestor

- mt-Eve: ~150000 years ago.
- Y-Adam: ~~60000~~ ~150000 years ago.


= mutant offspring
Generation n
Generation n-1
time
Present day


A
C
C
T
MRCA
A
C
G
T
A
C
C
T
A
C
C
T
A
C
C
T
T
C
C
T
A
C
G
T
A
C
G
T
time
Present day

Population genetics

- How population genetics influences the genetic variation data we see.
- Standard model: the coalescent.
- Coalescent Theory: An Introduction by John Wakeley.
- Hein, Schierup and Wiuf “Gene Genealogies, Variation and Evolution” - Oxford Press.

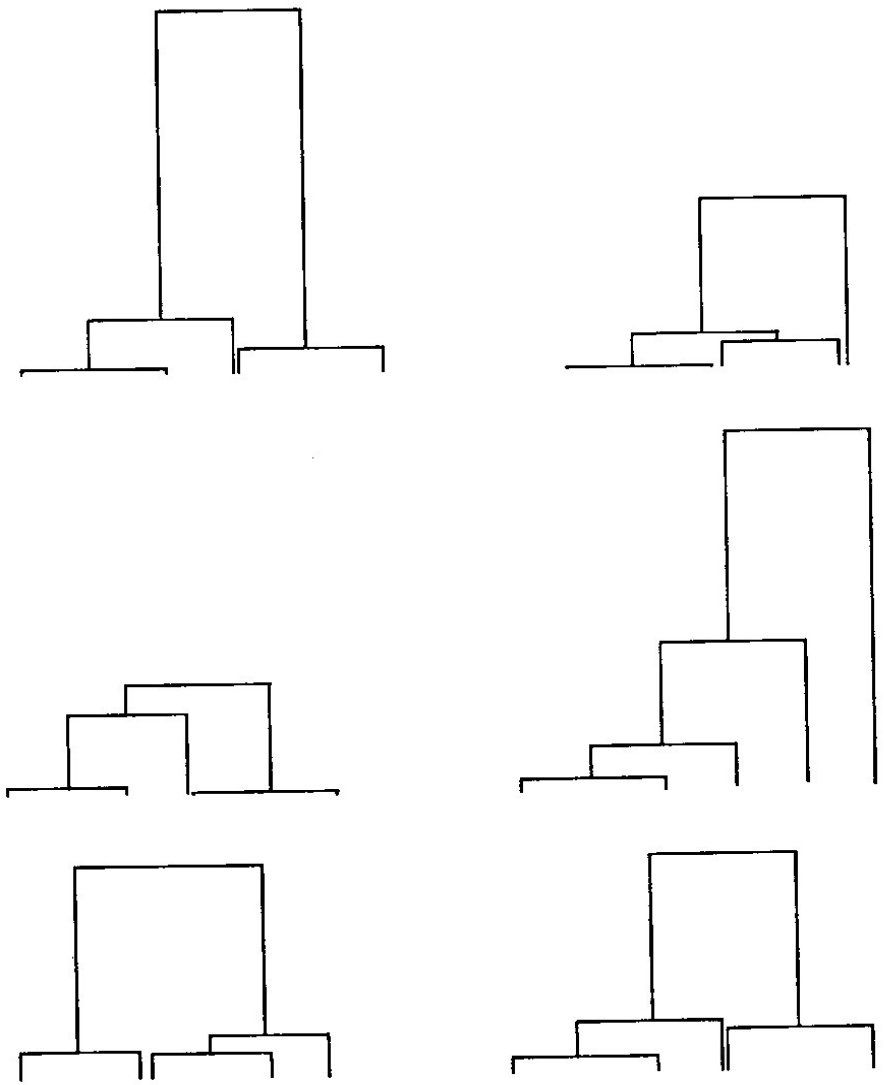


Fig. 2.3. Six realizations, drawn on the same scale, of coalescent trees for a sample of $n = 5$. (In each tree the labels 1,2,3,4,5 should be assigned at random to the leaves.)

Tavaré et al.

- Define S_n to be the number of segregating sites in a sample of size n
- Review a variety of results for samples of size 2
  - e.g. $(TMRCA|S_2=k)\sim\Gamma(1+k,1+\theta)$
  - $E(TMRCA|S_2=k)=(1+k)/(1+\theta)$
- Critique other papers (e.g. Templeton) that claim to use those results

- No useful theoretical results for more than 2 indivs, so use rejection method
- Summary stat S is # seg. sites
- S depends only on L (the length of the tree), so some theory is possible...

Note that we are using a summary statistic, S, rather than the full data. So this can also be seen as something called "Approximate Bayesian Computation" which we will meet later.

Recall Bayes theorem: $f(\theta|D) = \pi(\theta)f(D|\theta)/f(D)$

$\theta/2$ is the mutation rate parameter

$$f(t \mid S = k) \propto \int_0^\infty f(t,l)\mathrm{Po}(k, l\theta/2)dl,$$

where *f(t,l)* is the density of the tree height (t) and length (*l*) under the coalescent model. *Po(k,λ)* is the prob. of Poisson(*λ*) taking the value k.

So, the rejection method works as follows:

Rejection method

Let $P_o(k,\lambda)$ denote Pr(Poisson(λ)=k). Let $\theta/2$ be the mutation rate.

1. Generate a tree from the coalescent prior. Let t denote the TMRCA (height) for that tree, and let ℓ denote its length.
2. Accept the tree, and t, with probability $P_o(k, \ell\theta/2)$.
3. Return to 1.

Result: Independent samples from f(t | k segregating sites)
(NB., We are assuming the 'infinite sites' mutation model.)

Note that $P_o(k, \ell\theta/2)$ may be very small, so improve efficiency by using upper bound $P_o(k,k)$, where $P_o(k,k)= \max_\theta P_o(k, \ell\theta/2)$. So step 2. becomes:

2'. Accept tree, and t, with prob. $P_o(k, \ell\theta/2)/P_o(k,k)$.

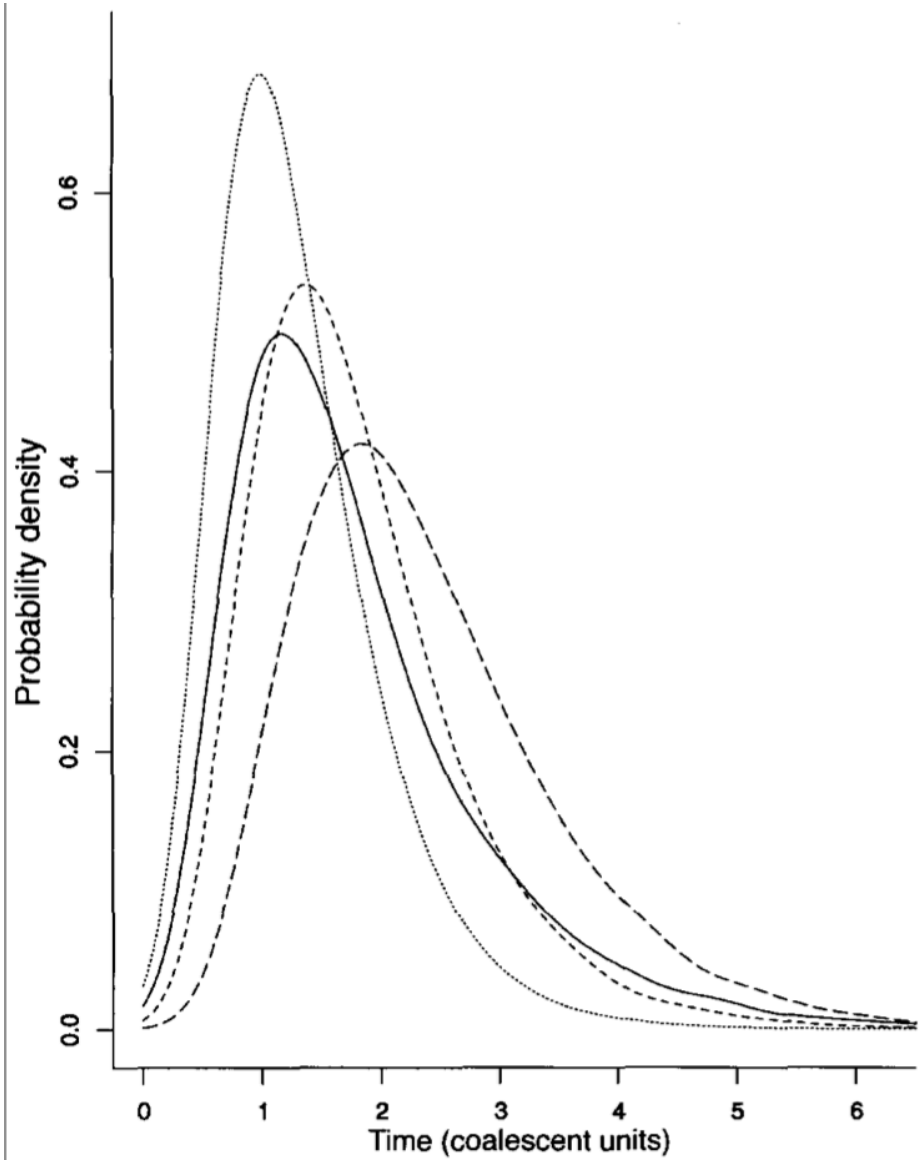


FIGURE 2.—Pre- and post-data density curves for T_{10} with $\theta = 1$. —, pre-data density; · · ·, $S_{10} = 1$; ---, $S_{10} = 3$; – – –, $S_{10} = 5$.

Results

TABLE 1

Effect of uncertainty about θ

	$\theta = 1$		θ random, $\mathbb{E}(\theta) = 1$	
Data	Mean of T_{10}	Variance of T_{10}	Mean of T_{10}	Variance of T_{10}
None	1.80	1.16	1.80	1.16
$S_{10} = 1$	1.30	0.44	1.60	0.87
$S_{10} = 3$	1.79	0.75	1.78	1.04
$S_{10} = 5$	2.38	1.19	1.90	1.18

Pre-data and post-data moments of T_{10} for three values of S_{10}, for θ known and θ random. The pre-data values are exact, from (2) and (3), whereas the post-data values are estimated from 10,000 iterations of Algorithm 1 or Algorithm 2. For θ random, its value is $2N\mu$, where N is lognormal (9,1) and μ is gamma with shape parameter 2 and mean 1/26,719. It follows that θ has mean 1, median 0.48 and SD 1.75.

Rejection method 2 - uncertainty regarding θ=2Nμ

1. Sample pop. size N from π_N
2. Sample mut^n rate μ from π_μ
3. Generate tree from coalescent prior
4. Accept with prob. Po(k, $N\mu L$)/Po(k,k) (in which case store T (or μ, *or* ...)
5. Return to 1.

Results

TABLE 1

Effect of uncertainty about θ

	$\theta = 1$		θ random, $\mathbb{E}(\theta) = 1$	
Data	Mean of T_{10}	Variance of T_{10}	Mean of T_{10}	Variance of T_{10}
None	1.80	1.16	1.80	1.16
$S_{10} = 1$	1.30	0.44	1.60	0.87
$S_{10} = 3$	1.79	0.75	1.78	1.04
$S_{10} = 5$	2.38	1.19	1.90	1.18

Pre-data and post-data moments of T_{10} for three values of S_{10}, for θ known and θ random. The pre-data values are exact, from (2) and (3), whereas the post-data values are estimated from 10,000 iterations of Algorithm 1 or Algorithm 2. For θ random, its value is $2N\mu$, where N is lognormal (9,1) and μ is gamma with shape parameter 2 and mean 1/26,719. It follows that θ has mean 1, median 0.48 and SD 1.75.

Results

TABLE 2

Reanalyses of data of HAMMER

	Data	Model	Mean of T_{16} ($\times 10^3$)		95% Interval ($\times 10^3$)	
			Pre-data	Post-data	Pre-data	Post-data
(a)	$S_{16} = 3$	HAMMER (11)		188		51–411
(b)	$S_{16} = 3$	$N = 4900$ $\mu_S = 9.88 \times 10^{-5}$	184	173	56–460	62–377
(c)	Full data (not insert) $S_{16} = 4$	$N = 4900$ $\mu_S = 9.88 \times 10^{-5}$ $N = 4900$	184	172	56–460	65–341

- Hammer (1995) had collected sequence data for a 2.6kb region for 16 human Y-chromosomes.
- He had some logical flaws in his analysis. The results above reanalyze that data using the rejection method.
- Note: Even for such a small dataset, exact calculation of likelihoods was not possible.

Conclusion

- Can use rejection methods to sample from a function (or probability density).
- Can still do this when the function or density cannot be sampled from directly.
- Can even still do it when the function or density cannot even be calculated.
- Use of these kind of methods has grown in the era of Big (genomic) Data.
- Uncertainty over parameter values reduces the strength of conclusions.

Rejection methods: Urn example

Draw until you have n non-black balls

Rules:

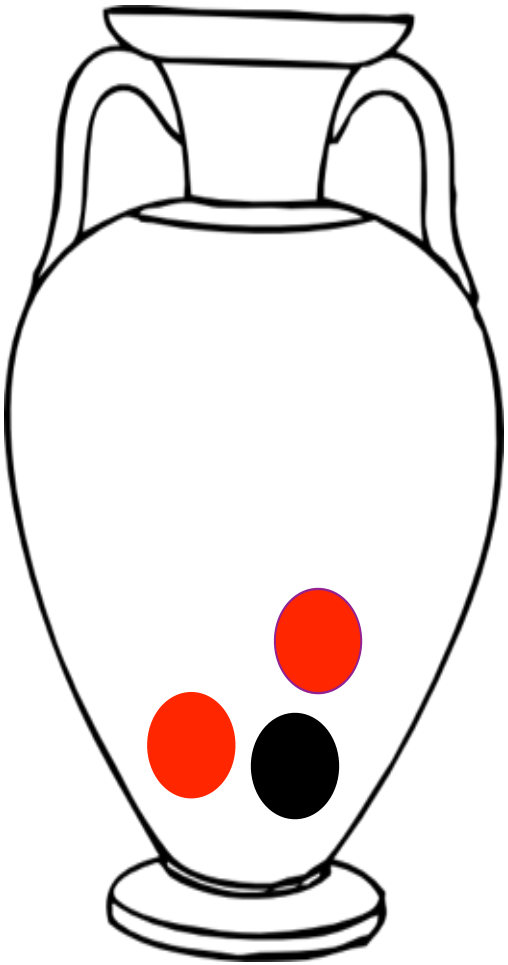
Draw black ball -> draw another ball, change its color, return both balls to the urn

Draw non-black ball -> return it to the urn along with another ball of the same color.

If there are currently m balls in the urn, we have

P(draw black ball) = w/(w+m-1)

P(draw colored ball) = m/(w+m-1)



Examinable assignment 4 - part 1

- Use the Urn model starting with 2 red balls of weight 1, and one black (mutation) ball of weight w.
- Draw balls until you have 10 non-black balls.
- If all non-black balls are the same color at the end, what is the posterior distribution of the weight of the black ball?
- If we observe exactly 2 non-black colors at the end, what is the posterior distribution of the weight of the black ball?
- Use a Uniform[0,20] prior for the weight of the black ball

Pseudocode

```
# Define a function Urn(m,n) that simulates an Urn that produces n non-black balls,
# assuming the black/mutation ball has weight m, and returns the number of non-black
# colors among the final n balls.  Then....

NoReps<-10000   # how many samples to generate
HowManyColorsNeeded<-1   # the 'target' we have to hit (what we saw in an observed dataset)
MaxWeight<-10    # the maximum weight we will consider for the black ball
NoOfBalls<- 10 # the number of non-black balls we want in the urn at the end
AcceptedWeights<-rep(-9,NoReps)

for (i in 1:NoReps){
    HowManyColorsObserved<- -9
    while (HowManyColorsObserved != HowManyColorsNeeded) {
          # Sample a weight, ThisWeight, for the mutation ball, from Unif(0,MaxWeight). Then...
          HowManyColorsObserved<-Urn(ThisWeight,NoOfBalls)  # simulate the urn using this weight
    }
   AcceptedWeights[i]<-ThisWeight
}

# you will now have NoReps accepted weights, so plot a histogram to see what the
# posterior distribution f(weight|HowManyColorsNeeded) looks like
```

[PseudoCode_Urn.R on Github Assignment 4]

END