

# **The statistical underpinnings of maximum likelihood and Bayesian inference**

Molecular Epidemiology of Infectious Diseases

Lecture 2

January 13th, 2020

# A word on likelihoods

A likelihood is the probability of **data  $X$**  given some **model  $M$**  and its **parameter values  $\theta$**

$$P(X|M, \theta)$$

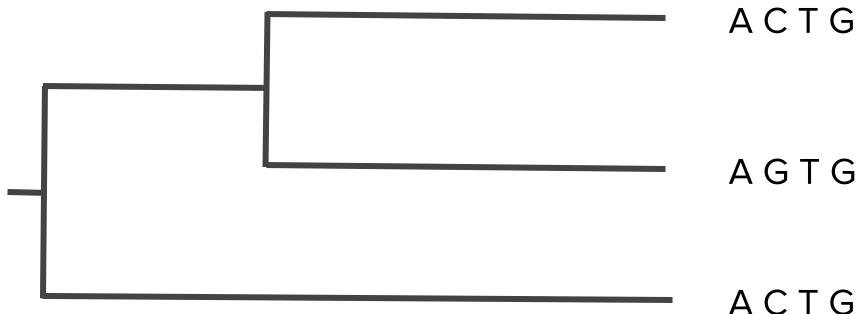
Likelihood based phylogenetic methods seek to find the tree that maximizes the likelihood of the sequence data under some model of molecular evolution

$$P(Seq|Tree, \theta)$$

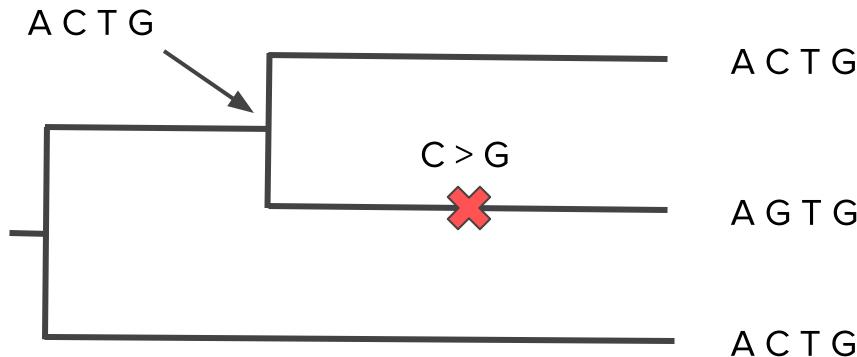
We therefore need to compute the likelihood of sequence data given a tree

**Let's start where we  
left off... assume we  
have a phylogeny  
with aligned  
sequences at the tips**

# Likelihood of sequence data on trees

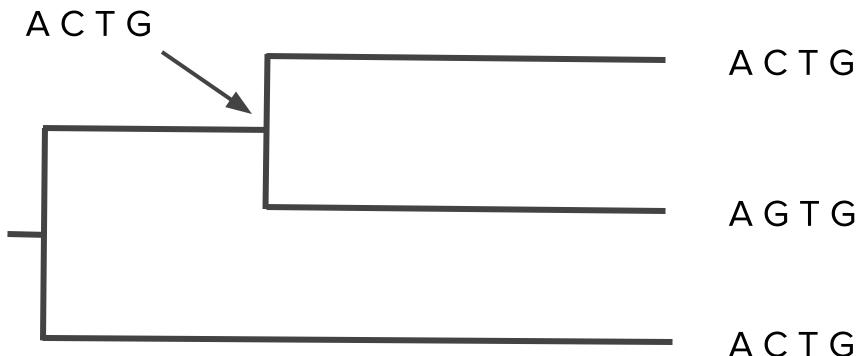


# Likelihood of sequence data on trees



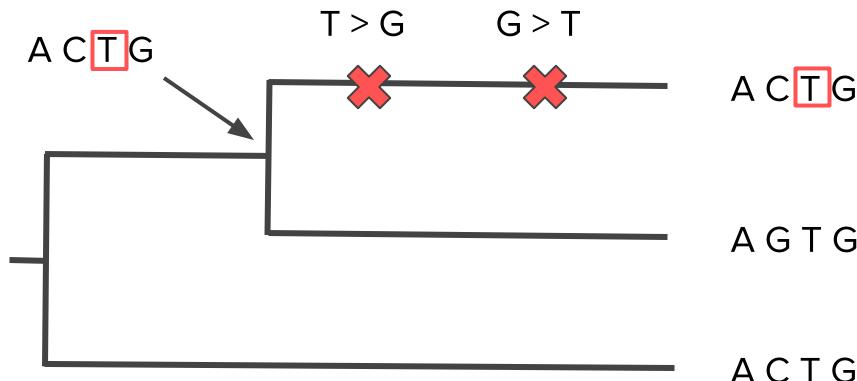
If we could directly observe sequence evolution on the tree, computing the likelihood of the sequence data would be easy. We could just compute the probability of every mutation event and multiply those probabilities together.

# Likelihood of sequence data on trees



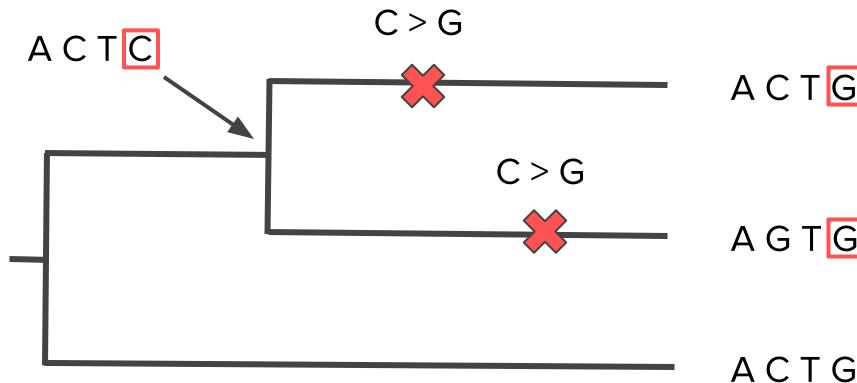
The problem is that we observe sequences at the tips but not their evolutionary history. Thus we have to take all possible evolutionary trajectories into account.

# Likelihood of sequence data on trees



This includes the possibility of **multiple substitutions** occurring at a particular site.

# Likelihood of sequence data on trees



And **convergent substitutions** occurring on different branches.

# Modeling molecular evolution

We normally model sequence evolution as a **Markov process**.

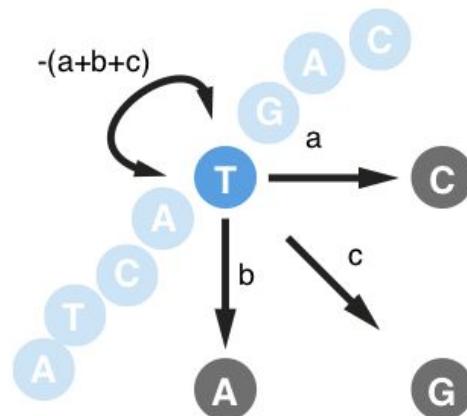
A Markov process is a type of **memoryless stochastic process**, i.e. a series of random experiments through time where the probability of jumping to a new state depends only the current state.

Example: the probability of a nucleotide base mutating to another base depends only on the current state, not previous states.

There are discrete and continuous time Markov processes. We generally model sequence evolution in continuous time.

# Markovian models of sequence evolution

At a given site, the rate at which transitions between different bases occur is given by a **substitution rate matrix**:

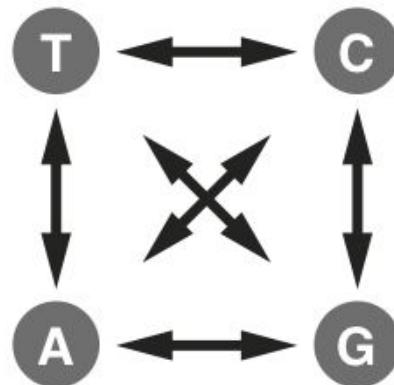


$$\begin{matrix} & \text{T} & \text{C} & \text{A} & \text{G} \\ \text{T} & -(a+b+c) & a & b & c \\ \text{C} & d & -(d+e+f) & e & f \\ \text{A} & g & h & -(g+h+i) & i \\ \text{G} & j & k & l & -(j+k+l) \end{matrix}$$

# **Some common substitution models for DNA sequence evolution**

# The Jukes-Cantor model

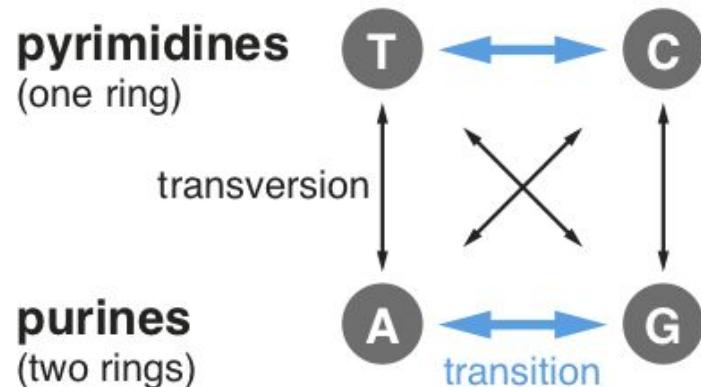
The Jukes-Cantor model is the most basic substitution model for nucleotide sequences. All substitutions have the same rate  $\lambda$ :



$$\begin{matrix} & T & C & A & G \\ T & \cdot & \lambda & \lambda & \lambda \\ C & \lambda & \cdot & \lambda & \lambda \\ A & \lambda & \lambda & \cdot & \lambda \\ G & \lambda & \lambda & \lambda & \cdot \end{matrix}$$

# The K80 model

The K80 model allows for two substitution rates, one for **transitions ( $\alpha$ )** and one for **transversions ( $\beta$ )**:

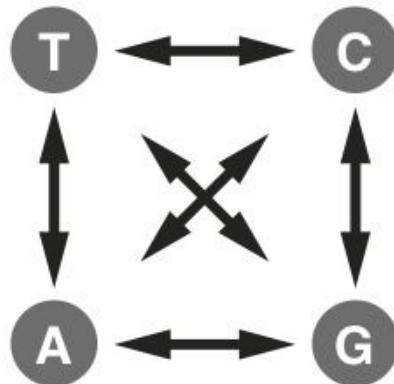


	T	C	A	G
T	.	$\alpha$	$\beta$	$\beta$
C	$\alpha$	.	$\beta$	$\beta$
A	$\beta$	$\beta$	.	$\alpha$
G	$\beta$	$\beta$	$\alpha$	.

Kimura (1980)

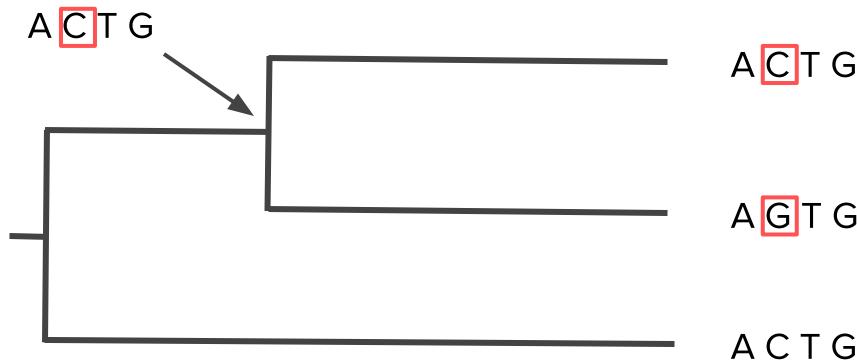
# The GTR model

The generalized time reversible model (GTR) allows for six different substitution rates for each pair of nucleotides but assumes rates are symmetric.



$$\begin{matrix} & \text{T} & \text{C} & \text{A} & \text{G} \\ \text{T} & \cdot & a\pi_C & b\pi_A & c\pi_G \\ \text{C} & a\pi_T & \cdot & d\pi_A & e\pi_G \\ \text{A} & b\pi_T & d\pi_C & \cdot & f\pi_G \\ \text{G} & c\pi_T & e\pi_C & f\pi_A & \cdot \end{matrix}$$

# Likelihood of sequence data on trees



So far we have rates of nucleotide substitutions, but we need to find **transition probabilities** to compute the likelihood.

# Modeling molecular evolution

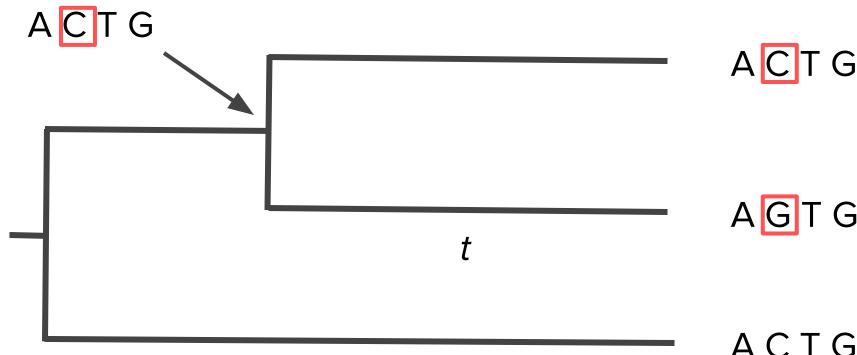
We can compute transition probabilities under a continuous-time Markov model given our substitution matrix  $\mathbf{Q}$  and the time elapsed along a branch  $t$ .

$$P(t) = e^{Qt}$$

The elements of  $P(t)$  give us the probability of every possible transition. Importantly, these **transition probabilities take into account every possible substitution path**.

$$P(t) = \begin{bmatrix} P_{T,T} & P_{T,C} & P_{T,A} & P_{T,G} \\ P_{C,T} & P_{C,C} & P_{C,A} & P_{C,G} \\ P_{A,T} & P_{A,C} & P_{A,A} & P_{A,G} \\ P_{G,T} & P_{G,C} & P_{G,A} & P_{G,G} \end{bmatrix}$$

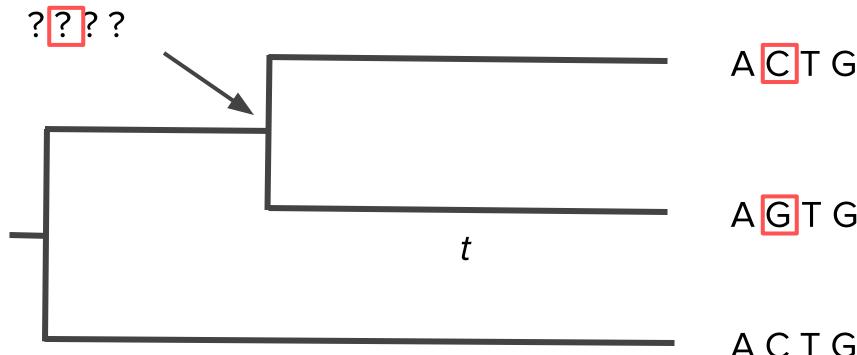
# Computing likelihoods at one site



Given the ancestral sequence of the parent, we can compute the likelihood at a single site:

$$L(Seq|Tree) = P_{C,C}(t) * P_{C,G}(t)$$

# Computing likelihoods at one site



If the ancestral sequences are not observed, we must integrate or sum over all possible ancestral states:

$$L(Seq|Tree) = \sum_{X \in \{A,C,T,G\}} (P_{X,C}(t) * P_{X,G}(t))$$

# Computing the total likelihood

**Felsenstein's pruning algorithm** (J. Mol. Evol., 1981) uses dynamic programming to compute likelihoods on larger trees. The algorithm traverses the tree from tips to root, combining the partial likelihoods of two subtrees at each internal node.

We generally assume sites evolve independently, so we can multiple the likelihood of each site to compute the total likelihood of the sequence data at all sites.

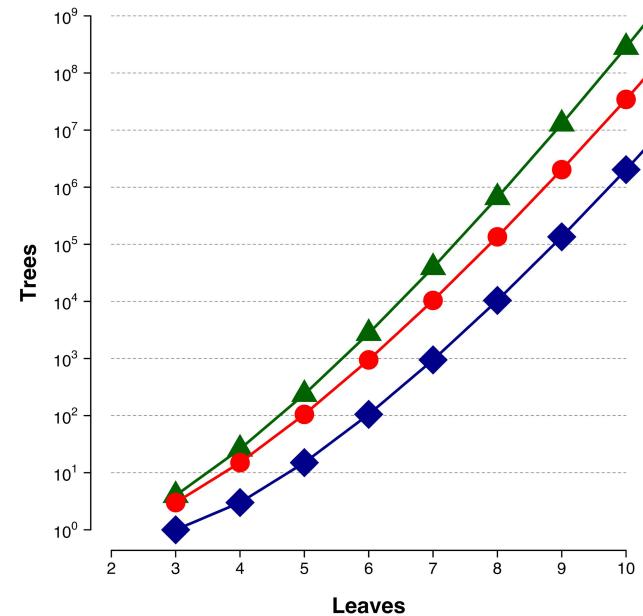
# Maximum likelihood tree reconstruction

Likelihood-based tree reconstruction methods **search tree space** to find the tree that maximizes the likelihood of the sequence data.

The number of potential trees grows rapidly with the number of tips. There are  $(2n-3)!!$  rooted binary trees for  $n$  tips.

Most ML methods like RAxML employ a heuristic rather than exhaustive tree searches.

\*\*\*Also need to estimate evolutionary parameters like substitution rates



Red shows rooted binary trees.

# Towards a Bayesian worldview

# Adopting a Bayesian worldview

Bayesian inference is really all about combining information in a rational way while dealing with uncertainty

**Basic model:** Prior beliefs → New data → Updated beliefs

The way we combine information follows directly from basic probability theory (i.e. Bayes theorem)

# Bayesian reasoning: An example

Let's say your doctor just diagnosed you with a very rare disease found in only one out of every 1,000 people (0.1% prevalence)

But you know the false positive rate of the diagnostic test is 5%

What is the probability that you are actually sick?

# Bayes theorem

Bayes theorem tells us how to correctly compute **conditional probabilities** of the form  $P(A|B)$ .

That is, what is the probability of observing outcome A given that we observed outcome B?

**Bayes theorem** tells us that:

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

# Bayes theorem: an example

In our example, we want to compute the conditional probability  $P(\text{sick} \mid +)$ .

Applying Bayes theorem, we see that:

$$P(\text{sick}|+) = \frac{P(+|\text{sick})}{P(+)} P(\text{sick})$$

# Bayes theorem: an example

We know that  $P(\text{sick})$  is 1 in 1000 = 0.001

And we now that the false positive rate is 5%, so  $P(+ | \text{healthy}) = 0.05$

We will assume that the false negative rate is also 5%, so that the true positive rate is  $P(+ | \text{sick}) = 0.95$ .

So the total probability of being diagnosed as positive is:

$$P(+) = P(+|\text{sick})P(\text{sick}) + P(+|\text{healthy})P(\text{healthy}) = 0.05$$

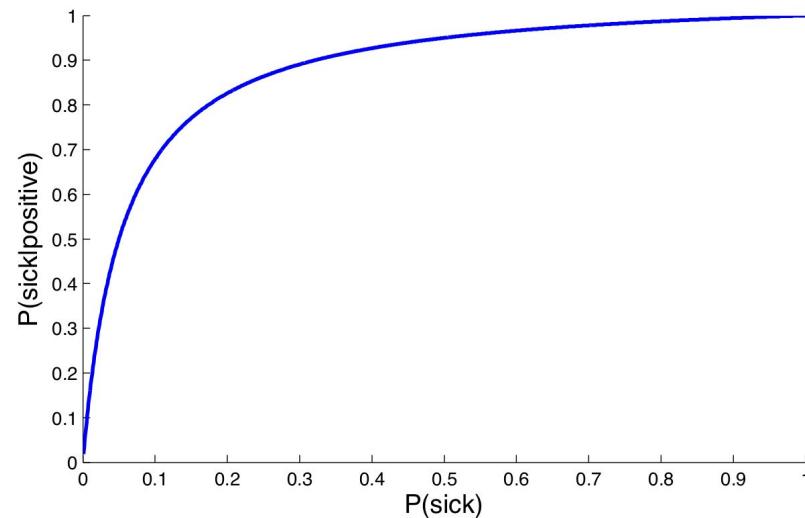
# Bayes theorem: an example

Putting everything back into Bayes theorem:

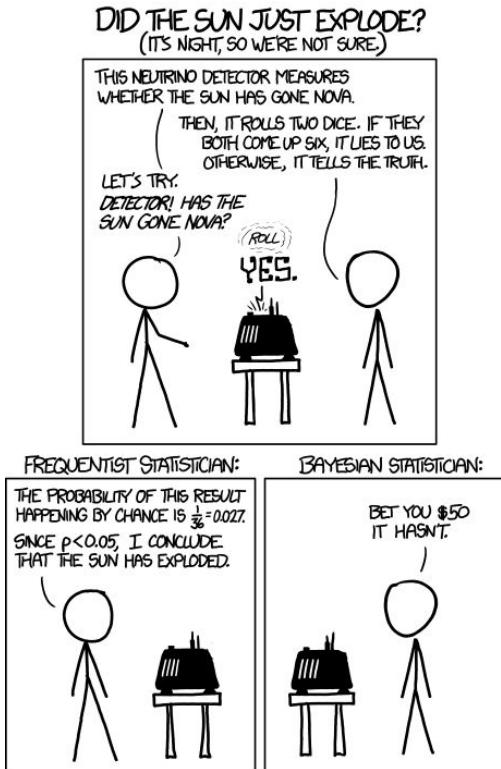
$$P(sick|+) = \frac{P(+|sick)}{P(+)} P(sick) = \frac{0.95}{0.05} 0.001 = 0.0187$$

# Dependence on prior beliefs

The actual probability of being sick given a positive test depends very strongly on the background rate or prevalence  $P(\text{sick})$ .



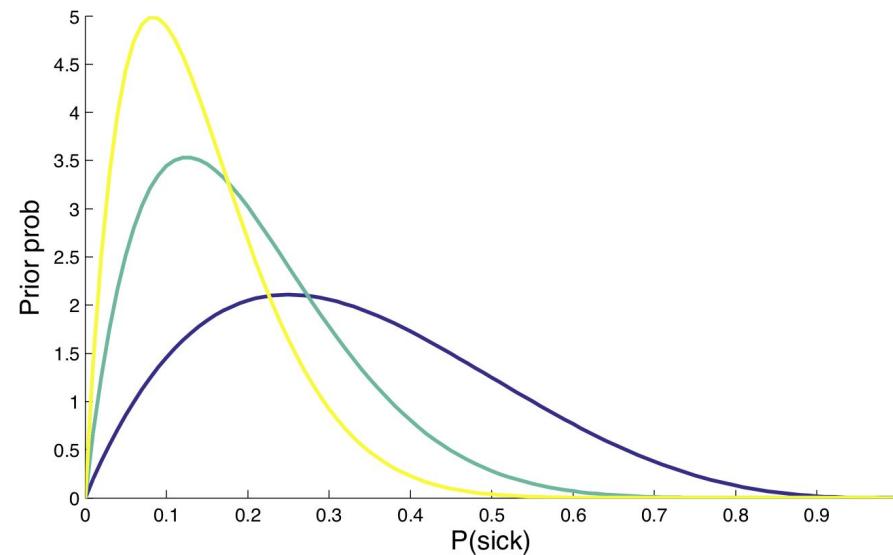
# The problem with ignoring prior info



# Bayesian priors

In Bayesian inference, we summarize our *a priori* beliefs using a prior distribution

The prior is a probability distribution over all possible values of an unknown (random) variable.



# Bayesian inference

Our prior beliefs get updated when we observe new information or data using Bayes theorem:

$$p(\theta|data) = \frac{L(data|\theta)}{p(data)} p(\theta)$$

# Bayesian inference

Our prior beliefs get updated when we observe new information or data using Bayes theorem:

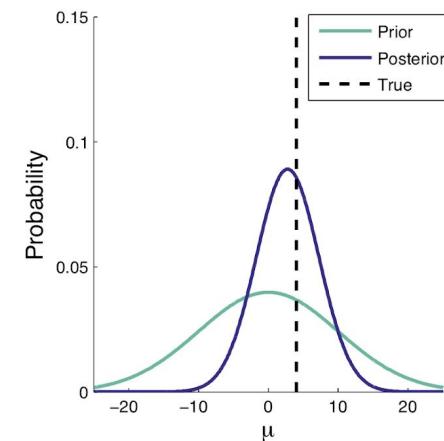
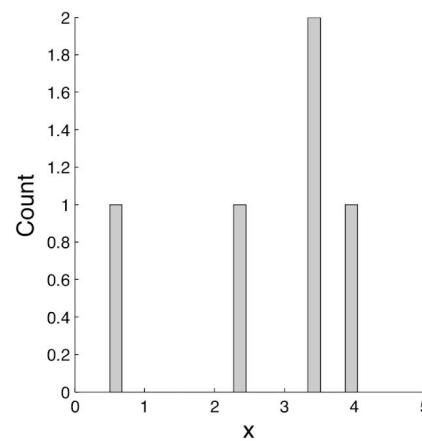
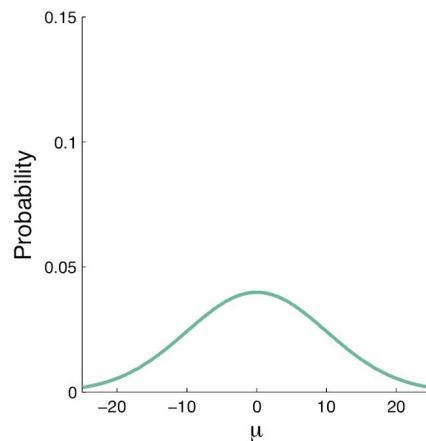
$$p(\theta|data) = \frac{L(data|\theta)}{p(data)} p(\theta)$$

The diagram illustrates the components of Bayes' theorem:

- Likelihood**: Represented by a downward arrow above the term  $L(data|\theta)$ .
- Prior distribution**: Represented by a diagonal arrow pointing towards the term  $p(\theta)$ .
- Posterior distribution**: Represented by a diagonal arrow pointing towards the term  $p(\theta|data)$ .
- Normalization constant**: Represented by an upward arrow below the term  $p(data)$ .

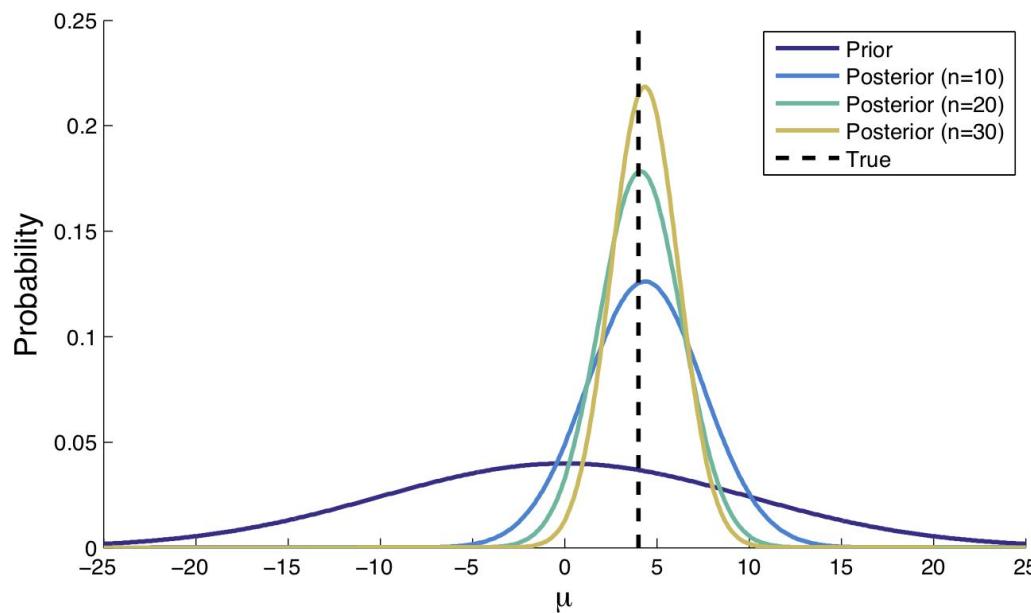
# Bayesian inference: another example

Inferring the mean value of a normally distributed population given a limited sample



# Bayesian inference: another example

The relative contribution of the prior to the posterior decreases with more data



# Computing the posterior

We can generally compute the unnormalized posterior probability of a given parameter value:

$$p(\theta = x | \text{data}) \propto L(\text{data} | \theta = x) p(\theta = x)$$

However, it is generally very difficult to compute the normalization constant:

$$p(\text{data}) = \sum_{\theta} L(\text{data} | \theta) p(\theta)$$

$$p(\text{data}) = \int_{\theta} L(\text{data} | \theta) p(\theta) d\theta$$

# Computing the posterior

If we cannot analytically compute the posterior, we can sample values from the posterior distribution and then use these samples to construct an approximation to the posterior distribution.

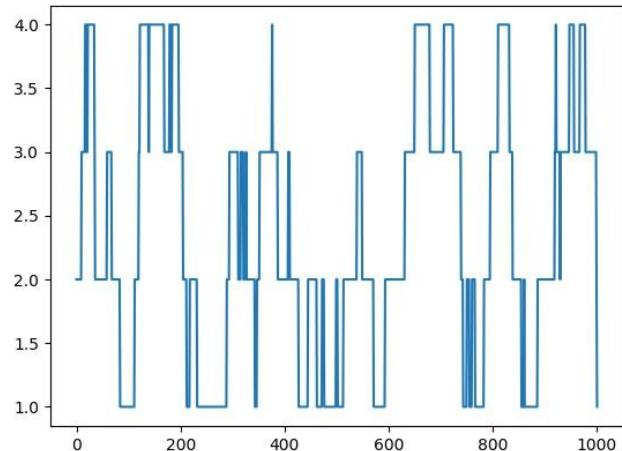
in Bayesian inference, **Markov chain Monte Carlo (MCMC)** is the most commonly used method to sample from a desired distribution.

# What is a Markov chain?

A Markov chain is a Markov process that randomly jumps between different states over time. The state of the process at time  $t_n$  depends only on the previous state at time  $t_{n-1}$ .

MCMC is an example of discrete-time process.

Example: a one dimensional random walk



# The Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm is one commonly used variant of MCMC:

At each MCMC iteration  $m$  with state  $x(m) = \theta$ :

1. Propose  $\theta^*$  from a proposal density  $q(\theta^*|\theta)$ .
2. Compute the acceptance probability  $\alpha$ :

$$\alpha = \frac{L(data|\theta^*)p(\theta^*)}{L(data|\theta)p(\theta)} \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}$$

3. If  $\alpha \geq 1$ : accept  $\theta^*$   
Else: accept  $\theta^*$  with probability  $\alpha$
4. If accepting  $\theta^*$ : set  $x(m + 1) = \theta^*$   
Else set  $x(m + 1) = \theta$ .

Hastings term

Ratio of posterior  
probabilities

# The Metropolis-Hastings algorithm

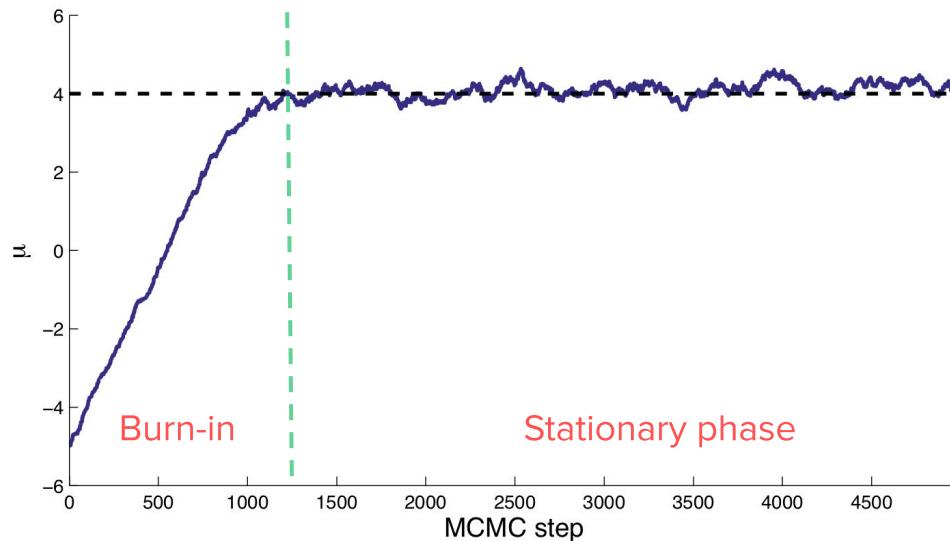
The **main idea behind the MH algorithm** is that we accept parameters with a probability proportional to their posterior probability.

This means that the amount of time the chain spends in state  $x$  will be proportional to the posterior probability of  $x$ .

However, for this to be true, the chain needs to have reached its **stationary phase or distribution** (i.e. equilibrium).

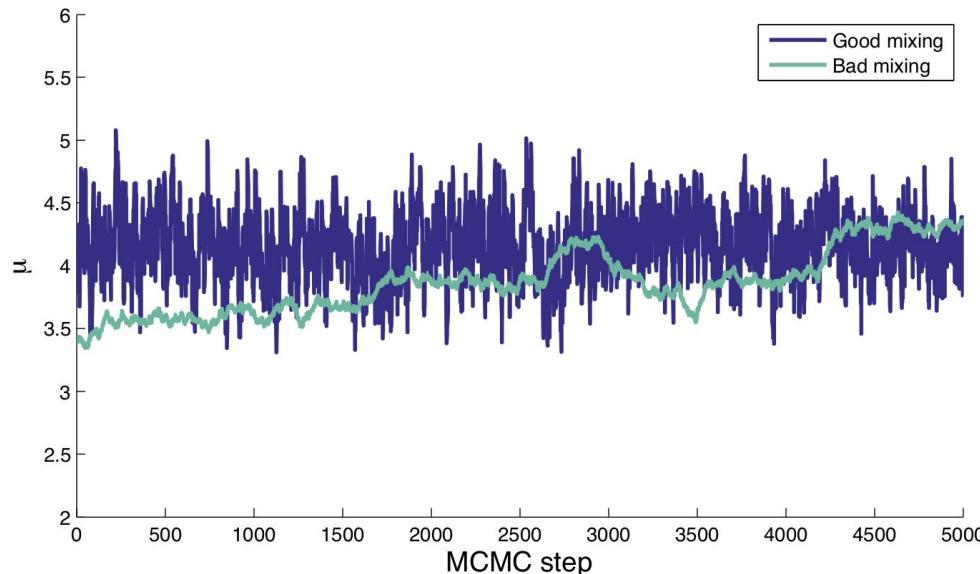
# MCMC: Convergence

Samples from a MCMC are only valid once the chain has **converged** on its stationary distribution



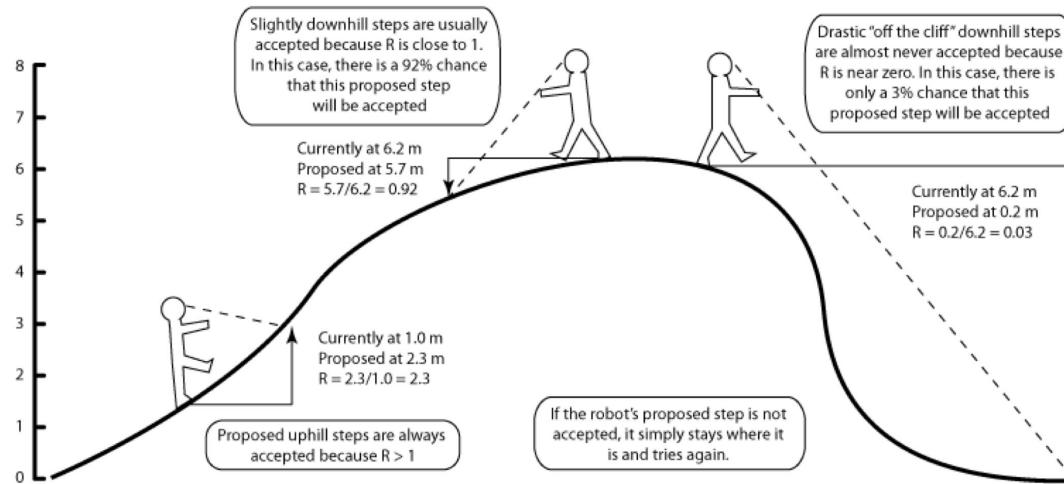
# MCMC: Mixing

**Mixing** refers to how efficiently the chain explores the posterior distribution.



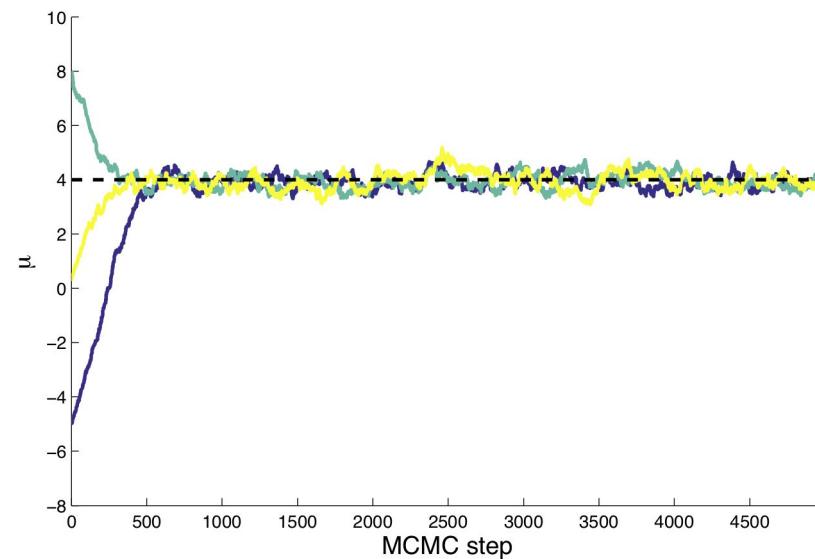
# MCMC: The blind robot analogy

Achieving good mixing requires a good proposal distribution.



# MCMC: Checking convergence

Because of issues with mixing and convergence, it is always a good idea to run multiple chains starting from different initial values.



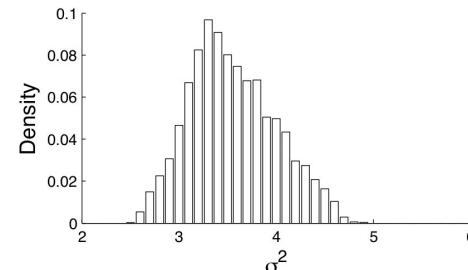
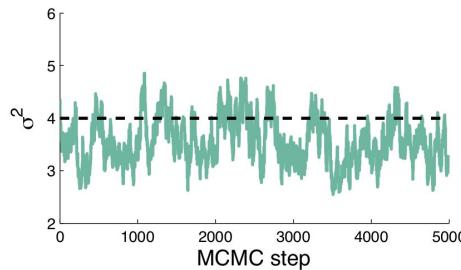
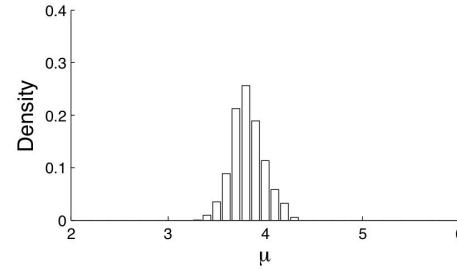
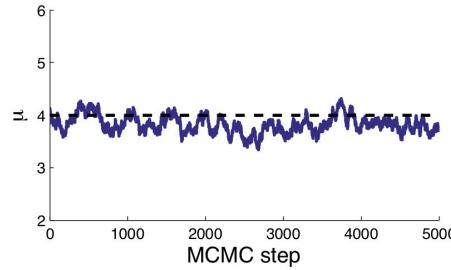
# MCMC in higher dimensions

MCMC is often used to infer the **joint posterior distribution** of two or more variables e.g.  $p(X, Y|Z)$

For many high-dimensional problems, MCMC is the only practical approach to Bayesian inference.

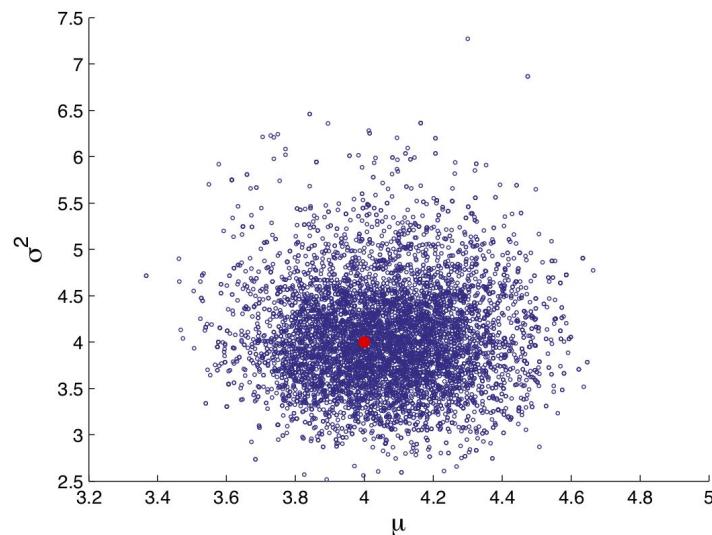
# MCMC: 2D example

Let's use the MH algorithm to estimate both the mean and variance of a normal distribution:



# MCMC: 2D example

In 2D, the amount of time the chain spends at a particular combination of parameters is proportional to their joint posterior probability.

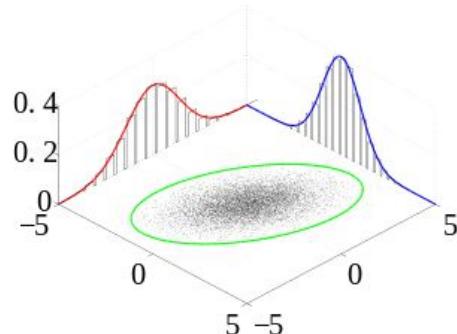


# Joint vs. marginal distributions

The **joint posterior** is the probability distribution over all unknown variables or parameters.

The **marginal posterior** is the probability distribution over a given parameter integrated (i.e. averaged) over all possible values of the other parameters.

Computing the marginal distribution allows us to take into account uncertainty in other estimated parameters.



# Summary of Bayesian inference

Bayes theorem tells us how to compute conditional probabilities of the form  $P(A|B)$  given we have information about  $P(A)$ .  $P(A)$  represents our prior beliefs about A.

Bayes theorem lets us compute the posterior distribution of a variable by combining prior information with new information coming from the data through the likelihood function.

Both the posterior and the prior are probability distributions over an unobserved (random) variable.

For many problems, we cannot directly compute the posterior but we can approximate it using MCMC.

# Bayesian phylogenetics

**“Have patience with everything  
that remains unsolved in your  
heart... live in the question.”**

**-Rainer Maria Rilke**

# From ML to Bayesian phylogenetics

Maximum likelihood (ML) methods of tree reconstruction focus on finding the tree that maximizes the likelihood of the sequence data given some model of molecular evolution.

But a single best tree can be misleading in that there may be a large number of alternative trees that can explain the data nearly equally well.

In theory then, we would like to consider a “forest” of likely trees.

# The Bayesian approach

Recall the general approach to Bayesian inference:

$$p(\theta|data) = \frac{L(data|\theta)}{p(data)} p(\theta)$$

The goal of Bayesian phylogenetics is to compute the posterior distribution of trees given a sequence alignment.

$$P(\mathcal{T}|Seq)$$

# Tree space

The **posterior tree distribution** is a probability distribution over the forest of all possible trees in tree space.

Tree space has a discrete component in that there are a finite number of possible **tree topologies** for a given number of taxa. But the number of possible topologies is typically huge.

Tree space also has a continuous component in that each branch has an associated length. There is therefore an infinite number of possible trees.

# MCMC in tree space

We can approximate the posterior tree distribution by sampling a large number of trees from the posterior using MCMC. While tree space is complex, the basic idea is very similar to other MCMC algorithms like Metropolis-Hastings.

At each MCMC iteration  $m$  with state  $x(m) = T$ :

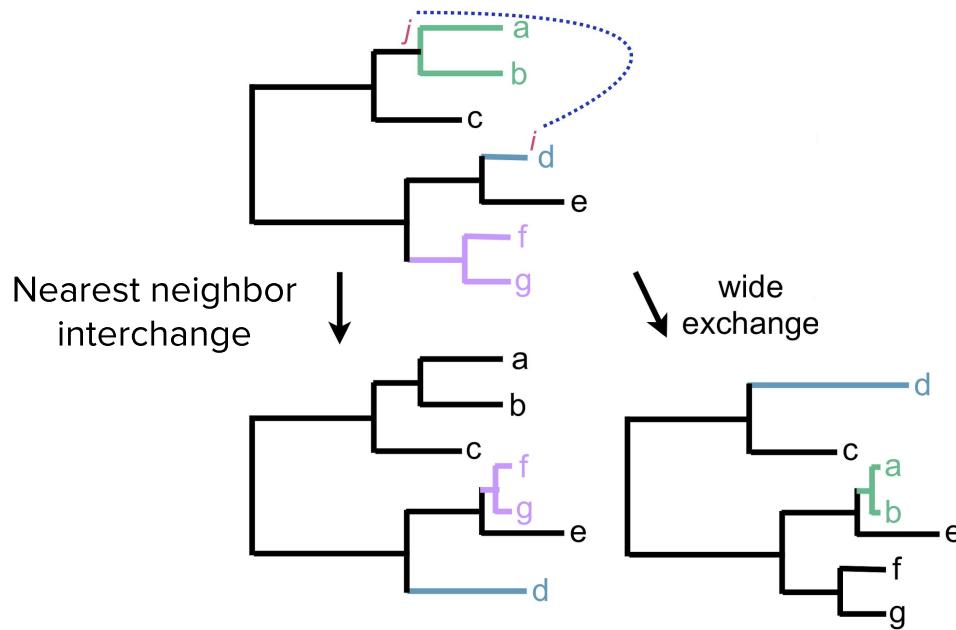
1. Propose  $T^*$  from a proposal density  $q(T^*|T)$ .
2. Compute the acceptance probability  $\alpha$ :

$$\alpha = \frac{L(\text{data}|T^*)p(T^*)}{L(\text{data}|T)p(T)} \frac{q(T|T^*)}{q(T^*|T)}$$

3. If  $\alpha \geq 1$ : accept  $T^*$   
Else: accept  $T^*$  with probability  $\alpha$
4. If accepting  $T^*$ : set  $x(m + 1) = T^*$   
Else set  $x(m + 1) = T$ .

# MCMC in tree space

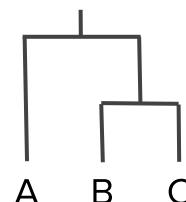
New tree topologies are proposed by rearranging the tree using subtree exchange.



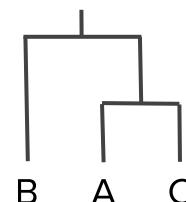
# Tree support

In Bayesian phylogenetics, the support for a given tree or bipartition is given by its posterior probability. Posterior probabilities are direct measure of our uncertainty.

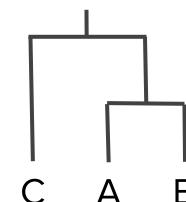
Using MCMC, the support or posterior probability for a given tree is approximated by its frequency in the posterior sample.



N=500  
P=0.5



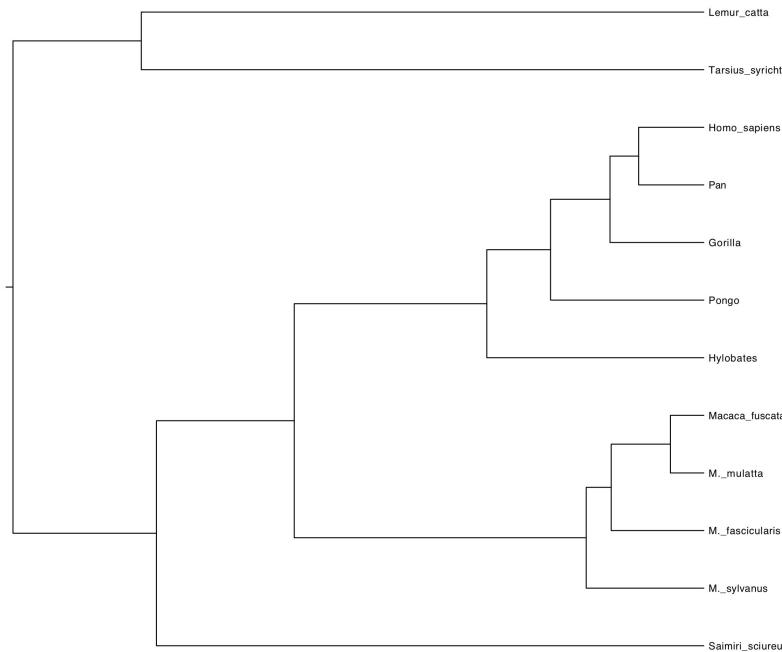
N=300  
P=0.3



N=200  
P=0.2

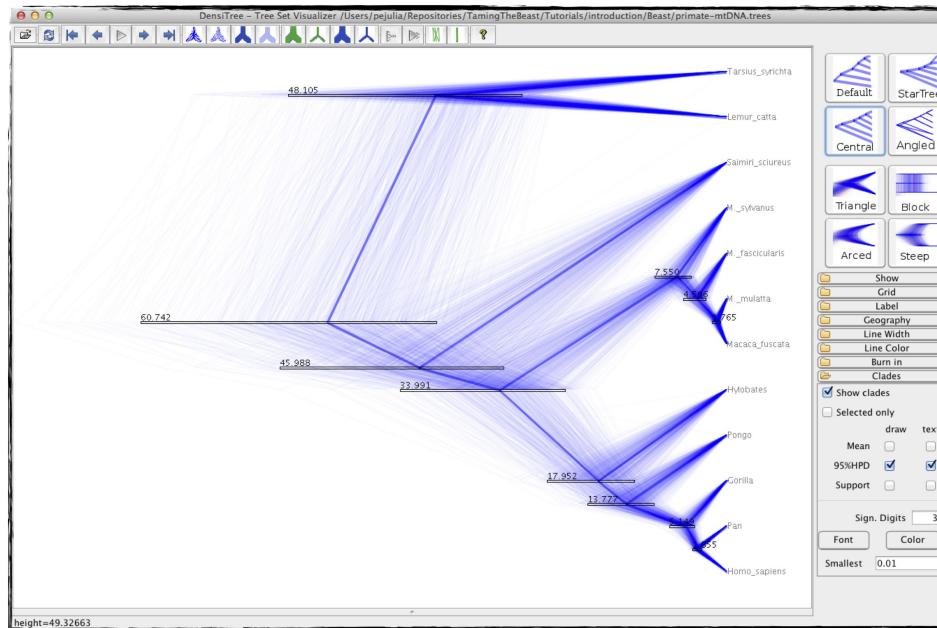
# Visualizing the tree posterior

See FigTree example for primate-mtDNA.trees



# Visualizing the tree posterior

DensiTree can also be used to overlay posterior tree samples.



# Now with all the moving parts

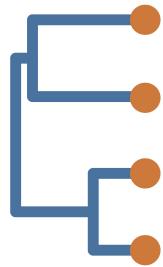
So far we have just considered the posterior tree distribution

$$P(\mathcal{T}|Seq)$$

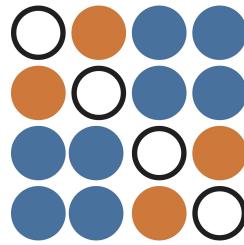
But typically there are several other parameters in the substitution model, molecular clock model and tree prior that must be jointly estimated together with the tree.

$$P(E \text{ } \square \square \text{ } \square | ACAC\dots) = \frac{P(ACAC\dots | E \text{ } \square \square \text{ } \square) P(E \text{ } \square \square \text{ } \square)}{P(ACAC\dots)}$$

# Now with all the moving parts



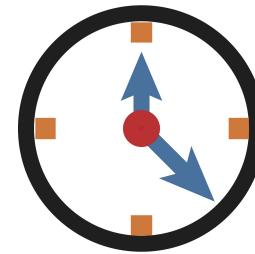
Tree



Substitution  
(Site) Model



Demographic  
Model (Tree  
Prior)



Molecular  
clock model

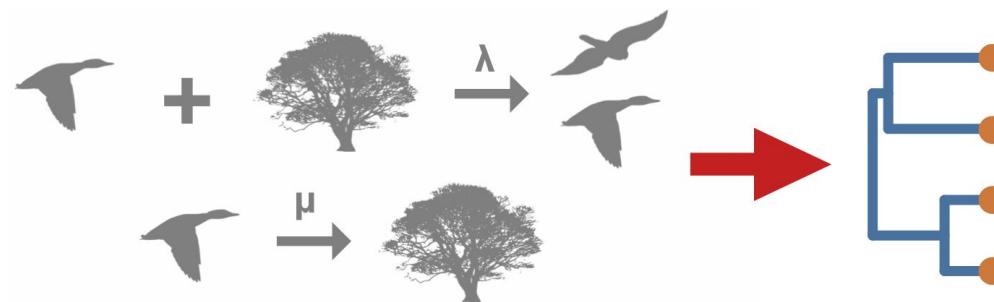


# Tree priors

In Bayesian phylogenetics, we need to place a prior distribution over tree space.

We could just assume that evolution is equally likely to produce any tree, resulting in a uniform prior over tree space.

But our tree prior should reflect our prior beliefs about the evolutionary processes that generated the tree (e.g. speciation and extinction rates).

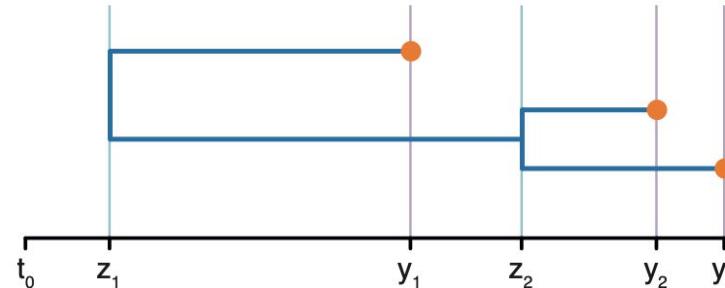




# Tree priors

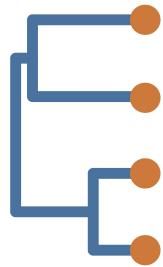
We will have several lectures on the *phylodynamic* models that are used as tree priors in molecular epidemiology, including coalescent and birth-death models.

**Birth-death →**

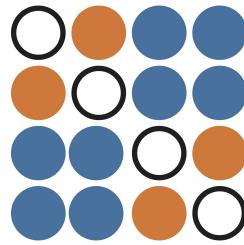


← **Coalescent**

# Now with all the moving parts



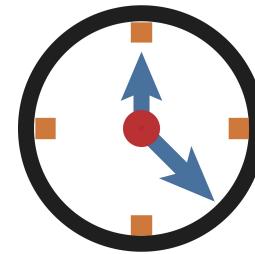
Tree



Substitution  
(Site) Model

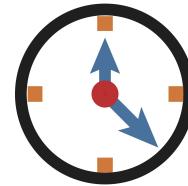


Demographic  
Model (Tree  
Prior)

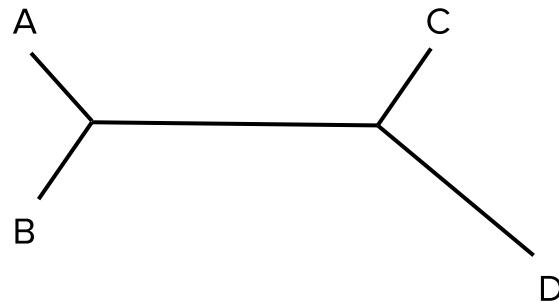


Molecular  
clock model

# The dating problem

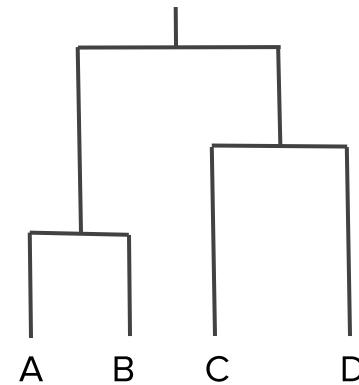
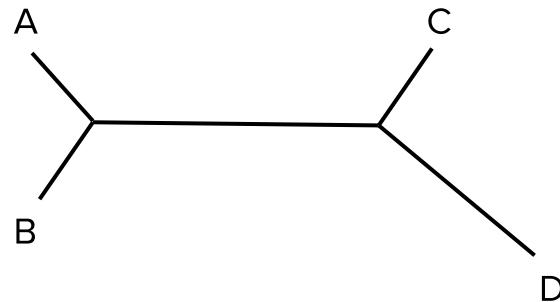


Most traditional (ML) phylogenetic methods infer unrooted trees that are not dated i.e. branch lengths are not time-calibrated in units of real time.



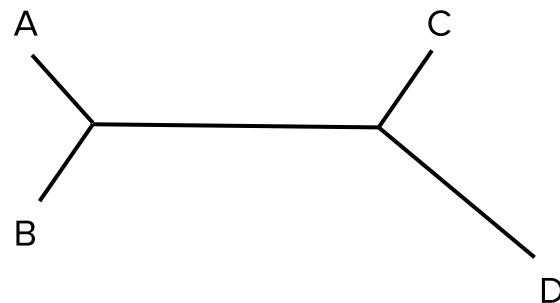
# The dating problem

In Bayesian phylogenetics we normally work with rooted trees.

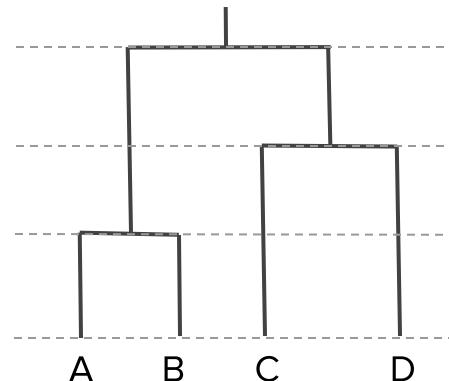


# The dating problem

Rooting the tree places additional constraints on node heights and branch lengths.



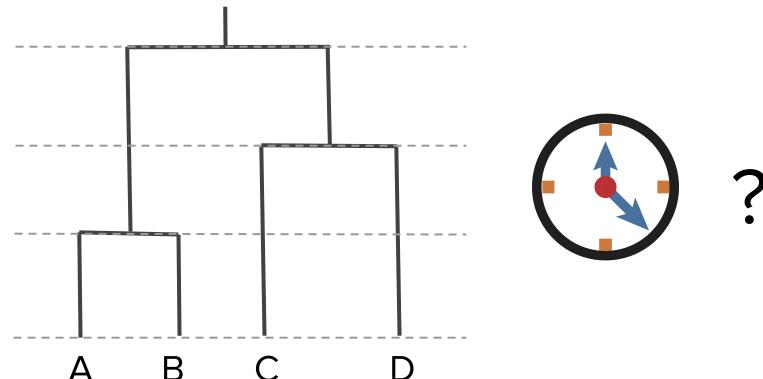
Branch lengths are unconstrained



Rooting the tree constrains branch lengths since all tips need to be equidistant to root

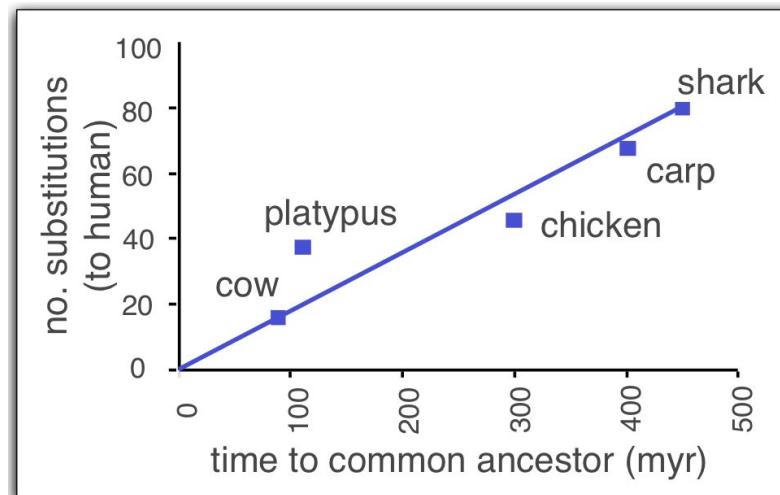
# The dating problem

But how do we assign dates to the nodes in the tree?



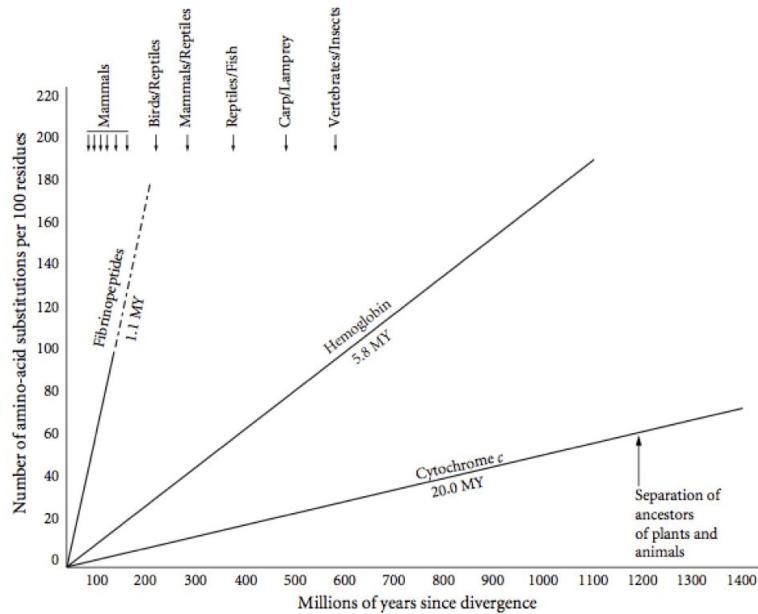
# The molecular clock

Zuckerkandl and Pauling (1962) found that the genetic distance between different animal haemoglobins increases almost linearly with the divergence time of the two species.



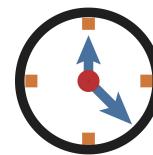
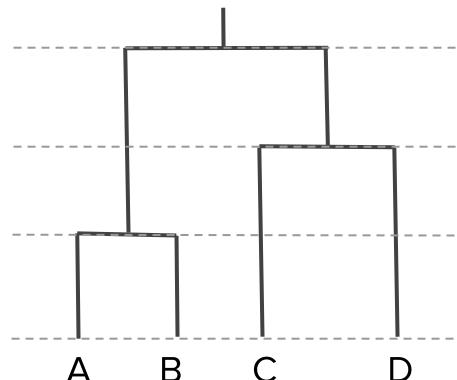
# The molecular clock

This pattern holds for many different proteins, implying that substitutions accumulate at a constant rate over time but at different rates in different proteins.



# The molecular clock model

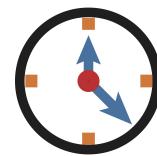
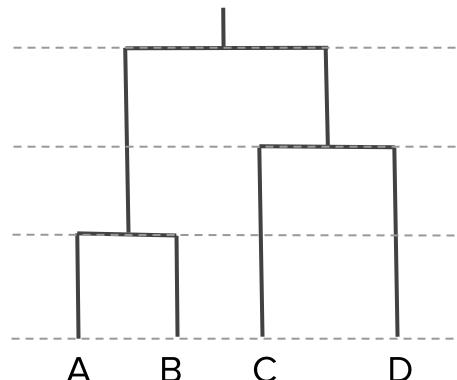
Genetic distance = **clock rate** X **time**



$$\text{Time} = \frac{\text{Genetic distance}}{\text{clock rate}}$$

# The molecular clock model

But we need additional information/constraints in order to estimate the clock rate from the phylogeny.



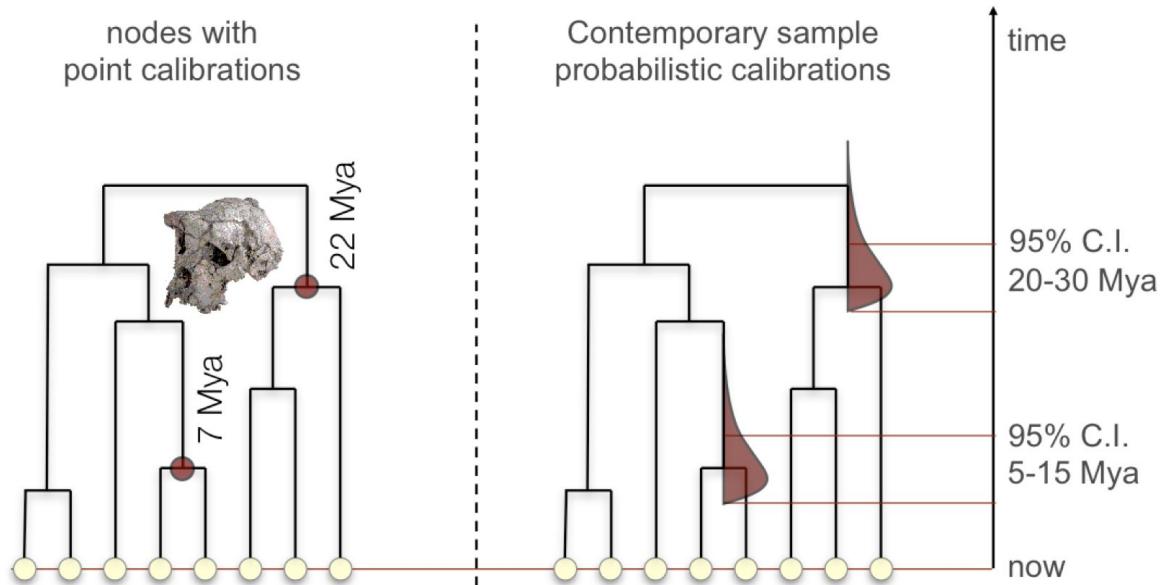
$$\text{Time} = \frac{\text{Genetic distance}}{\text{clock rate}}$$

# Two ways to calibrate the molecular clock

1. Time point calibrations on divergence times
2. Serially sampled (heterochronous) sequence data

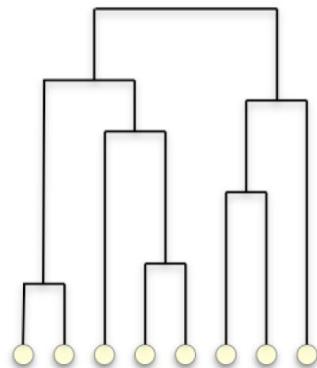
# Calibrating from divergence times

Specifying divergence times on particular nodes.

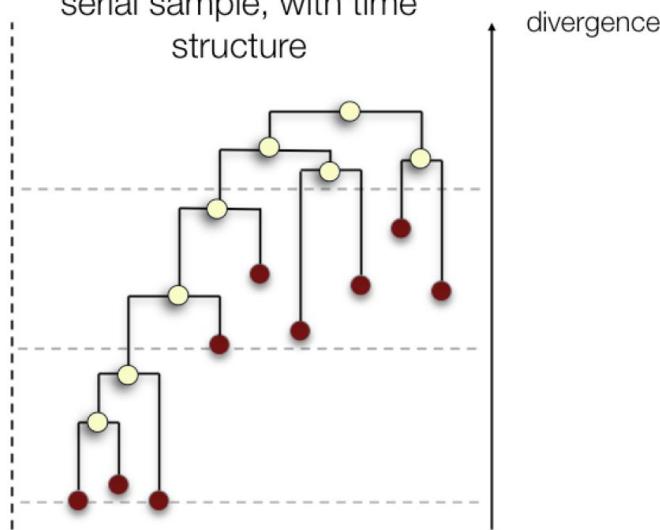


# Calibrating from serial tip times

contemporary sample,  
no time structure



serial sample, with time  
structure

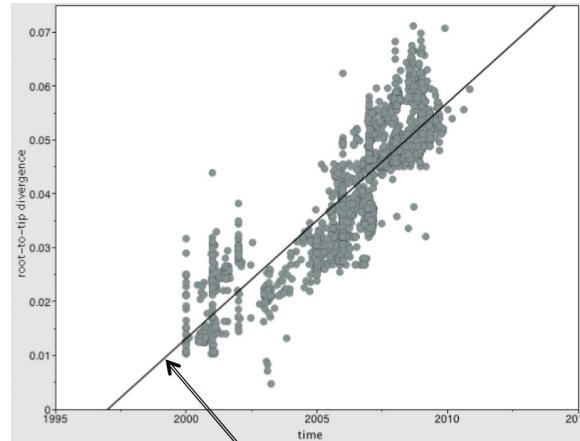


► Rambaut A. (2000) *Bioinformatics*, **16**, 395-399.

# Calibrating from serial tip times

In Bayesian phylogenetics, we jointly infer the clock rate and all node heights/branch lengths using MCMC. But the idea is conceptually similar to regressing root-to-tip genetic divergence against sampling times in terms of where the information is coming from. The slope is an estimate of the clock rate..

Influenza A H1N1  
2000-2011  
**Clock rate** =  $4.38 \times 10^{-3}$

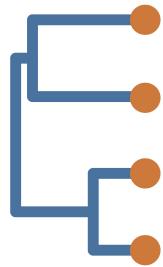


# Calibrating from serial tip times

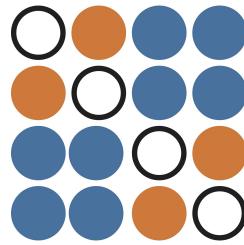
There must be sufficient time for mutations to occur between sampling events in order for the clock rate to be estimated.

The slower the mutation rate or the more closely together sequences are sampled in time the less information there will be about the clock rate.

# Now with all the moving parts



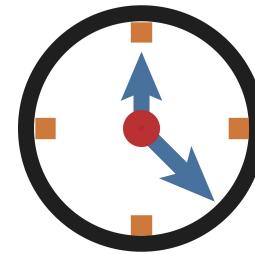
Tree



Substitution  
(Site) Model



Demographic  
Model (Tree  
Prior)



Molecular  
clock model

# Conclusion

Bayesian phylogenetic analysis can be complex with many interacting models each with their own set of parameters.

Bayesian phylogenetics lets us quantify uncertainty about these parameters and the phylogeny by inferring the posterior tree distribution using MCMC.

We may only be interested in the tree or specific parameters, but performing joint inference in a Bayesian setting allows us to take into account uncertainty about all parameters including the tree.

# Why Beast2?



**Beast2**

Bayesian evolutionary analysis by sampling trees

Implements many popular evolutionary and phylodynamic models. Plus many add-on packages.

Very efficient MCMC due to optimized proposals

Written in Java, runs everywhere.

Well-documented with lots of online community support