PM520: Lecture 7

Further explorations in MCMC

Examinable assignment 3 - part 1

- Use the IndeptGamma.R code to use MCMC to produce samples from a Gamma(2.3,2.7) random variable
- Show how the performance deteriorates when you run the algorithm to produce samples from a Gamma(0.1,0.01).
  - Use the Gelman plot to illustrate the deterioration in performance
- Find a proposal kernel (i.e. q(x->x'), the way of producing new candidate values) that performs more efficiently.
  - Again use the Gelman plots to show how performance has changed.
  - Discuss how the performance has improved

- Due on April 5th, at 1pm

Code on GitHub in Assignment3 as 'IndeptGamma.R'

Examinable Assignment 3 part 2: Code-breaking (due March 30th, 1pm)

- Gzo uclfg gcpo C qhcs okof te Gollk Qoeetb zo rhf slvem ce h LtqqfLtkio Fcqxol Rlhcgz tvgfcso gzo gollhio tu Gzo Sheiolf. Gzo ahlmced qtg hggoesheg zhs yltvdzg gzo ihl tvg hes zo rhf fgcqq ztqsced gzo sttl taoe yoihvfo Gollk Qoeetbf qoug uttg rhf fgcqq shedqced tvgfcso, hf cu zo zhs utldtggoe zo zhs teo. Zo zhs h ktvedqttmced uhio yvg zcf zhcl rhf yteo rzcgo. Ktv itvqs goqq yk zcf okof gzhg zo rhf aqhfgolos gt gzo zhclqceo, yvg tgzolrcfo zo qttmos qcmo hek tgzol ecio ktveddvk ce h sceeol jhimog rzt zhs yooe faoesced gtt pviz pteok ce h jtceg gzhg obcfgf utl gzhg avlatfo hes utl et tgzol. Gzolo rhf h dclq yofcso zcp. Zol zhcl rhf h qtxoqk fzhso tu shlm los hes fzo zhs h scfgheg fpcqo te zol qcaf hes txol zol fztvqsolf fzo zhs h yqvo pcem gzhg hqptfg phso gzo LtqqfLtkio qttm qcmo jvfg hetgzol hvgtptycqo. Cg scseg wvcgo. Etgzced ihe. Gzo hggoesheg rhf gzo vfvhq zhqugtvdz izhlhigol ce h rzcgo ithg rcgz gzo ehpo tu gzo lofghvlheg fgcgizos hiltff gzo ulteg tu cg ce los. Zo rhf doggced uos va. "Qttm, pcfgol," zo fhcs rcgz he osdo gt zcf xtcio, "rtvqs ktv pces h rztqo qtg avqqced ktvl qod cegt gzo ihl ft C ihe mces tu fzvg gzo sttl Tl fztvqs C taoe cg hqq gzo rhk ft ktv ihe uhqq tvg" Gzo dclq dhxo zcp h qttm rzciz tvdzg gt zhxo fgvim hg qohfg utvl ceizof tvg tu zcf yhim. Cg scseg ytgzol zcp oetvdz gt dcxo zcp gzo fzhmof. Hg Gzo Sheiolf gzok dog gzo ftlg tu aotaqo gzhg scfcqqvfcte ktv hytvg rzhg h qtg tu dtquced pteok ihe st utl gzo aolftehqcgk.

[CodedMessage_Short.txt in Assignment3 repo on Github]

Assignment 3 - Part 2

- Use one of the methods you have met in the course to try to break the code.
- I may not be easy to break the code completely without resorting to manual ‘tweaks’ at the end, but you should be able to get to the point at which you can work out what the text is saying.
- Write it up as an Rmarkdown file and include a description of the methods you are using (for both parts of the assignment) and a discussion of your results. Upload it to your version of the Assignment 3 repo on Github please.
- Due April 5th, 1pm.

Suggestion

- Need something to maximize, so let's maximize the likelihood.
- Suppose we have a sequence of letters $l_1\ l_2\ l_3\ l_4\ l_5$. Then [Recall $P(A|B)=P(A\cap B)/P(B)$]:
  - $P(l_1\ l_2\ l_3\ l_4\ l_5)=P(l_1)\ P(l_2|l_1)\ P(l_3|l_2\ l_1)\ P(l_4|l_3\ l_2\ l_1)\ P(l_5|l_4\ l_3\ l_2\ l_1)$
- Suppose the sequence of letters is Markovian. Then:
  - $P(l_1\ l_2\ l_3\ l_4\ l_5)=P(l_1)\ P(l_2|l_1)\ P(l_3|l_2)\ P(l_4|l_3)\ P(l_5|l_4)$
  - $P(l_1\ l_2\ l_3\ l_4\ l_5)=P(l_1)\ [P(l_1l_2)/P(l_1)]\ [P(l_2l_3)/P(l_2)]\ [P(l_3l_4)/P(l_3)]\ [P(l_4l_5)/P(l_4)]$
- Let $f_{\alpha\beta}$ denote the frequency with which the letter pair $\alpha\beta$ is observed in text (from the file on Blackboard). Let f_α denote the frequency with which the letter α is observed in text. Then $f_\alpha = \sum_\beta f_{\alpha\beta}$.
- Use $f_{\alpha\beta}$ as an estimate of $P(\alpha\beta)$ and f_α as an estimate of $P(\alpha)$.
- Now treat it as a maximization (of likelihood) or MCMC problem.

Non examinable exercise 2

- Code up the Metropolis-Hastings MCMC algorithm for sized-biased sampling from an arbitrary distribution over the range [0,10].

  – Test your algorithm by using it to construct sized-biased samples from an exp(λ) distribution. Plot your results and compare them to the curve $x\lambda e^{-\lambda x}$.

  – Recall: Sized-biased samples from a density f(x) have density xf(x), rather than f(x)

  – Note that your algorithm will work for any density function f.

Pseudocode on Github in SizeBiasedMCMC

Gelman stationarity test

- Run two (or more) replicate analyses and compare the answers.
- Compare the variance of the parameter distribution in the two (or more) chains.
- If both chains have reached stationarity, the variance within each replicate analysis will be the same as that across (i.e. after combining) the two replicates.
- So the Gelman test compares “across run” variance to “within run” variance (dividing the former by the latter). Stationarity has been reach if the statistic is close to 1 (“less than 1.1” is a common “rule of thumb”).

MCMC: Metropolis-Hastings

1. If at *x*, propose move to *x′* according to transition kernel q(*x* -> *x′*).
2. Calculate

$$h = \min\left\{1, \frac{f(x')q(x' \to x)}{f(x)q(x \to x')}\right\}$$

3. Move to *x'* with prob. h, else remain at *x*.
4. Return to 1.

(Metropolis et al. 1953, Hastings 1970)

The Markov chain has stationary distribution f.

Metropolis-Hastings: Features

- Pros:
  - The algorithm ‘learns’.
  - Comparison is ‘local’.
  - MCMC package in R, Gelman or CODA convergence diagnostics, WinBUGS, Many others…

- Cons:
  - Need to be able to calculate f(x) (may be non-trivial xin some cases - we will see an example of this today).
  - Need to sample from stationary distribution.
  - Consecutive outputs are correlated.
  - May get stuck in local minima

- Book chapter on Blackboard “Chapter on MCMC” - read up to section 7.4.3 (inclusive)

Sampling from Posterior Distributions

- Data, D; model parameter(s) θ. Density f(D|θ).

- Bayes theorem:

$$f(\theta|D) = f(D|\theta)\pi(\theta)/f(D)$$

Likelihood

Prior distribution

Normalizing constant

MCMC: Metropolis-Hastings

1. If at θ, propose move to θ' according to transition kernel q(θ -> θ').
2. Calculate

$$h = \min\left\{1, \frac{[P(\theta'|D)q(\theta' \to \theta)}{[P(\theta|D)q(\theta \to \theta')}\right\}$$

3. Move to θ' with prob. h, else remain at θ.
4. Return to 1.

The stationary distribution of this chain will be P(θ | D)

MCMC: Metropolis-Hastings

1. If at θ, propose move to θ' according to transition kernel $q(\theta \to \theta')$.
2. Calculate

$$h = \min\left\{1, \frac{[P(D|\theta')\pi(\theta')/P(D)]q(\theta' \to \theta)}{[P(D|\theta)\pi(\theta)/P(D)]q(\theta \to \theta')}\right\}$$

3. Move to θ' with prob. h, else remain at θ.
4. Return to 1.

The stationary distribution of this chain will be $P(\theta \mid D)$

MCMC: Metropolis-Hastings

1. If at θ, propose move to θ' according to transition kernel q(θ -> θ').
2. Calculate

$$h = \min\left\{1, \frac{P(D|\theta')\pi(\theta')q(\theta' \to \theta)}{P(D|\theta)\pi(\theta)q(\theta \to \theta')}\right\}$$

3. Move to θ' with prob. h, else remain at θ.
4. Return to 1.

The stationary distribution of this chain will be P(θ | D).

P(D) disappears, **so this can be used even when *P(D)* cannot be calculated!**

Example MCMC Application

- Wilson and Balding: Genealogical inference from microsatellite data, Genetics 150:499-510, 1998.

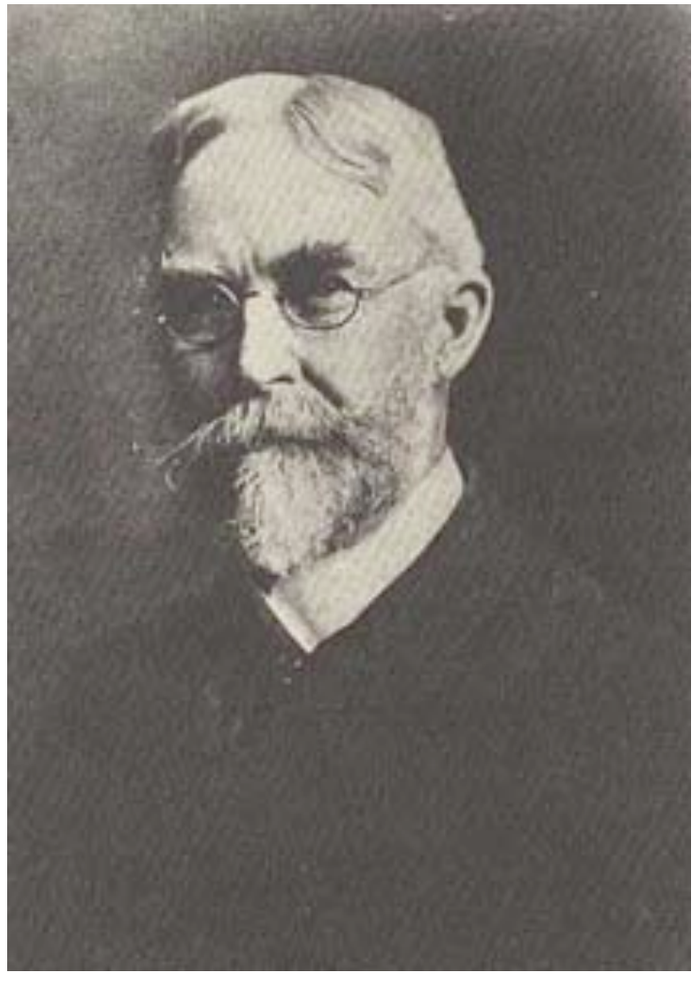
- Outline:
  - Use micro-satellite (short tandem repeats) data for pop. gen. evolution
  - Good: easy to collect
  - Bad: hard to interpret (*back mutation*)
  - Use MCMC on coalescent (ancestral) trees ***with an augmented state-space.***

Micro-satellite

- Number of repeats varies from person-to-person
- Used as markers in population studies
- Very common; mostly neutral but some associated with disease (Trinucleotide repeat disorders, e.g. Fragile X syndrome, Huntington's disease)
- Mutation rate relatively high (slippage)
-

Huntington's disease

- Described by George Huntington in 1872
- CAG repeat on short arm of chr 4
- Neurodegenerative
- More repeats -> earlier onset
- Number of repeats is most likely to increases via paternal inheritance

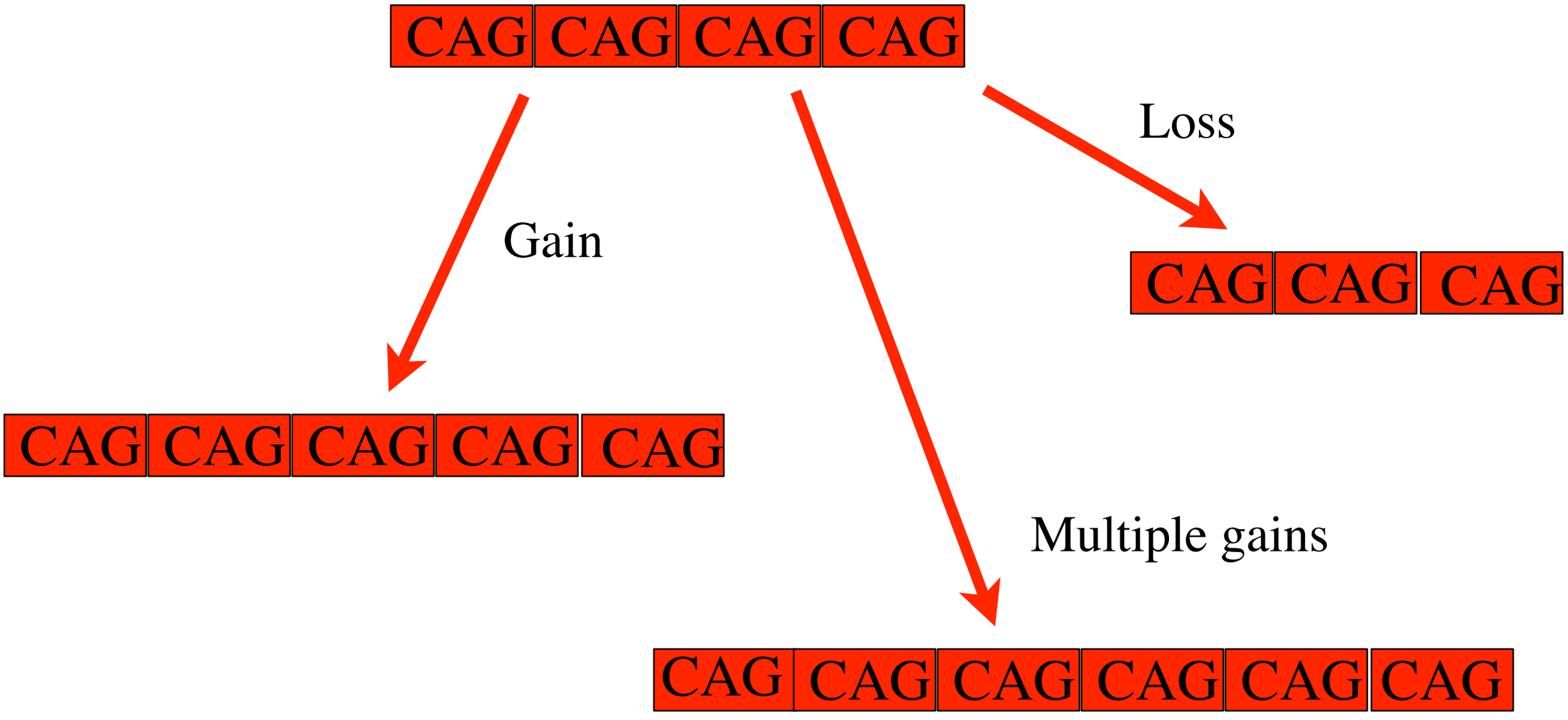


Classification of the trinucleotide repeat, and resulting disease status, depends on the number of CAG repeats[25]

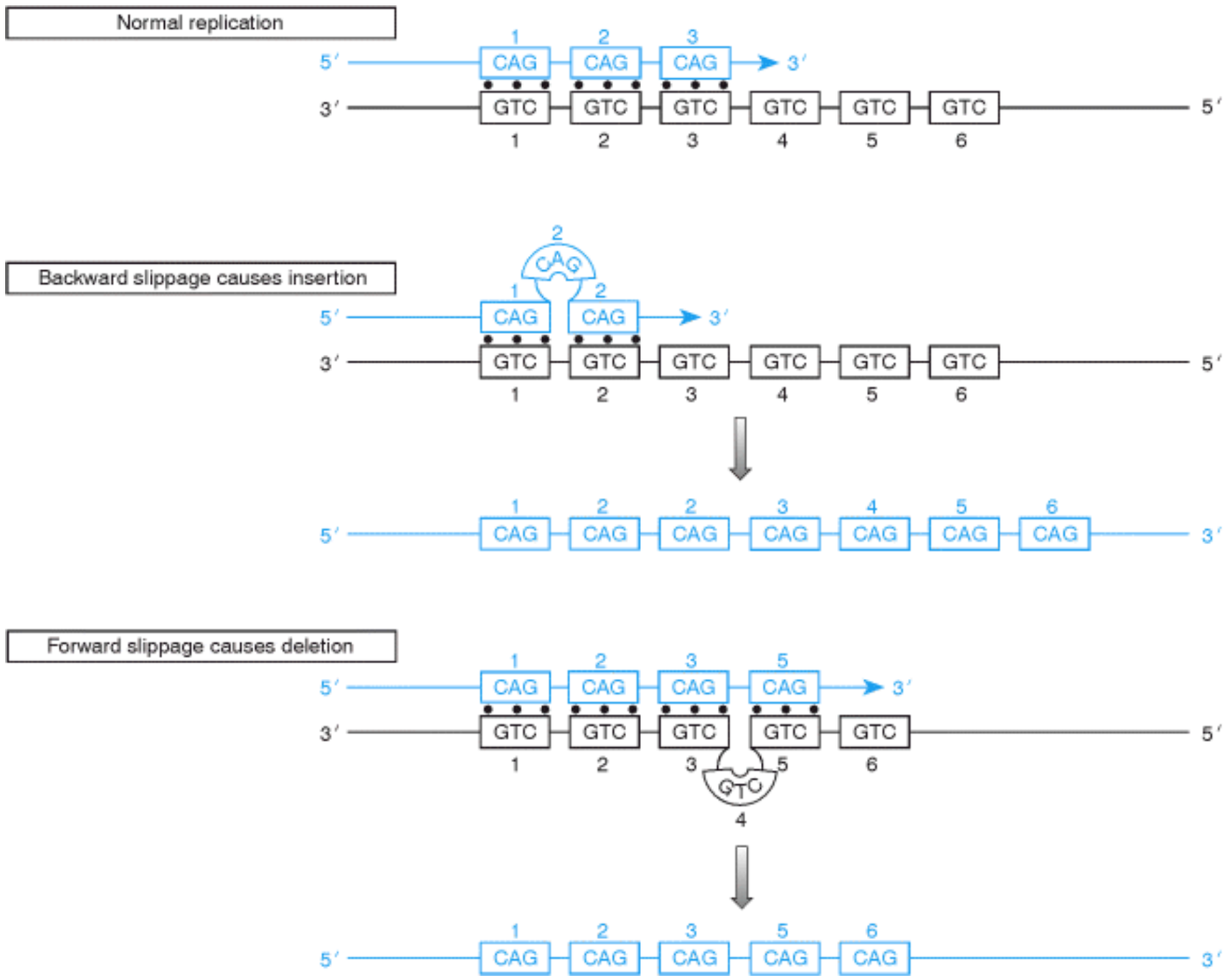
Repeat count	Classification	Disease status	Risk to offspring
<26	Normal	Will not be affected	None
27–35	Intermediate	Will not be affected	Elevated but <<50%
36–39	Reduced Penetrance	May or may not be affected	50%
40+	Full Penetrance	Will be affected	50%

https://en.wikipedia.org/wiki/Huntington%27s_disease

Micro-satellite mutation



Slipped-strand mispairing



http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=hmg.figgrp.1097

- Back-mutation is very possible (i.e. not every ‘mutation’ will lead to a state never seen before)

- Step-wise (‘ladder’) mutation models:
  - #repeats changes by +/- 1. (Weber and Wong 1993, Heyer et al, 1997, observed 11 mutations, all of which were +/-1)

- Mutation rate may depend on #repeats (but not in this paper)

Background

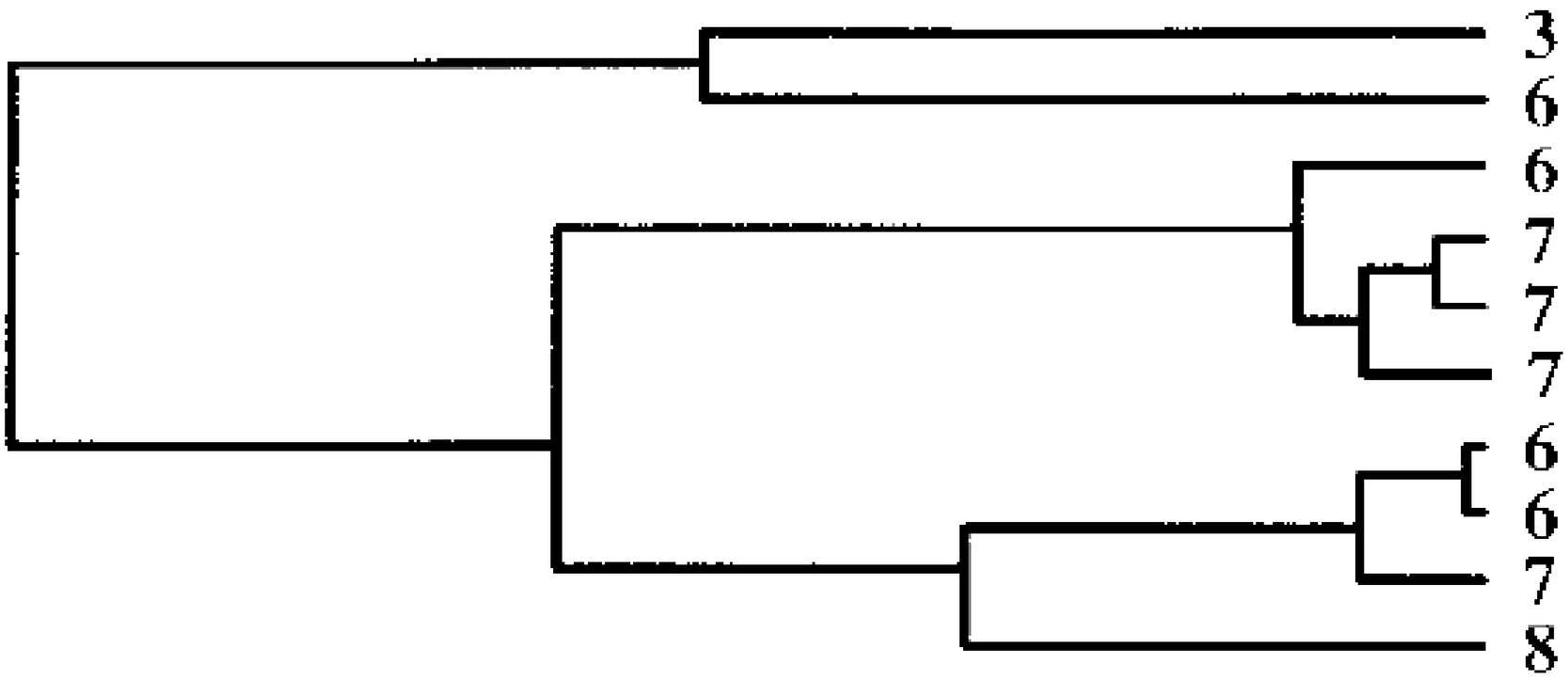
- Traditionally, pop. gen. inference had been based on things such as pairwise differences (Manhattan distance) - i.e., simple things.
- But, pairwise summaries of the data miss much of the info (e.g., genealogy)
- Methods based on ancestral trees were appearing (Felsenstein et al., Tavaré and Griffiths), but these methods ignored back-mutation.

- Data, D:
  - collection of samples from individuals.
  - each sample consists of an observation of the number of repeats at a set of L micro-satellite loci.
- Parameter:
  - Mutation parameter, θ.
  - Model: (Unobserved) ancestral tree showing how the samples are related to each other + action of mutation.

- Goal: to infer the mutation rate.
- Problem: Direct calculation of f(θ|D) is impossible, as is calculation of f(D).
- Solution: Use MCMC.

$$f(\theta|D) = f(D|\theta)\pi(\theta)/f(D)$$

- MCMC algorithm: explores the space of trees that explain the data



(From Figure 1 of the paper)

MCMC

$$f(\theta|D) = f(D|\theta)\pi(\theta)/f(D)$$

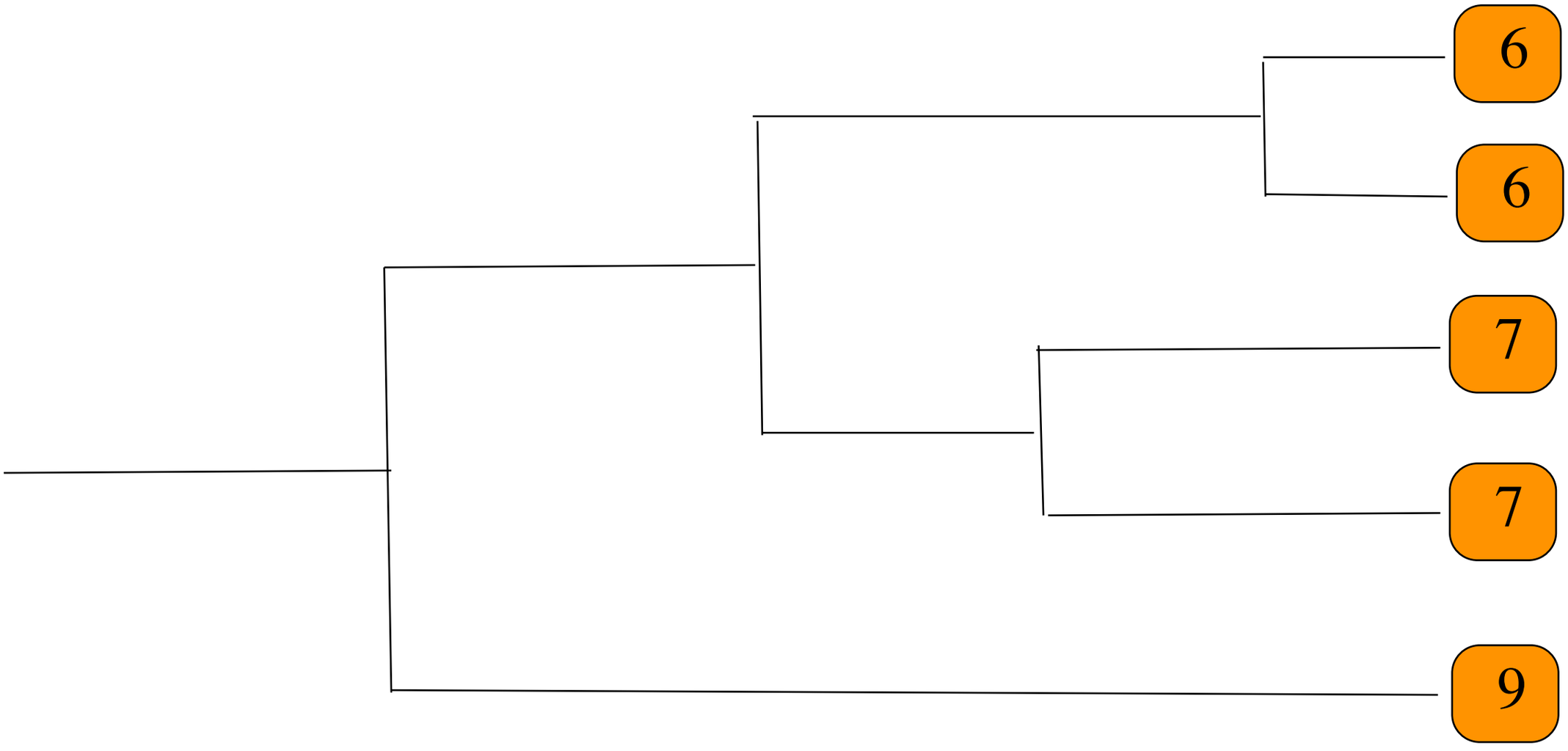
- Use of MCMC requires calculation of P(D|θ), the prob. of the data given the mutation rate, and a good way of moving around the parameter-space (i.e. a good proposal kernel, q).

- But, calculation of P(D|θ) directly is impossible, so use an *augmented state-space* that also includes the unobserved coalescent tree of the data: let Y denote the current tree.

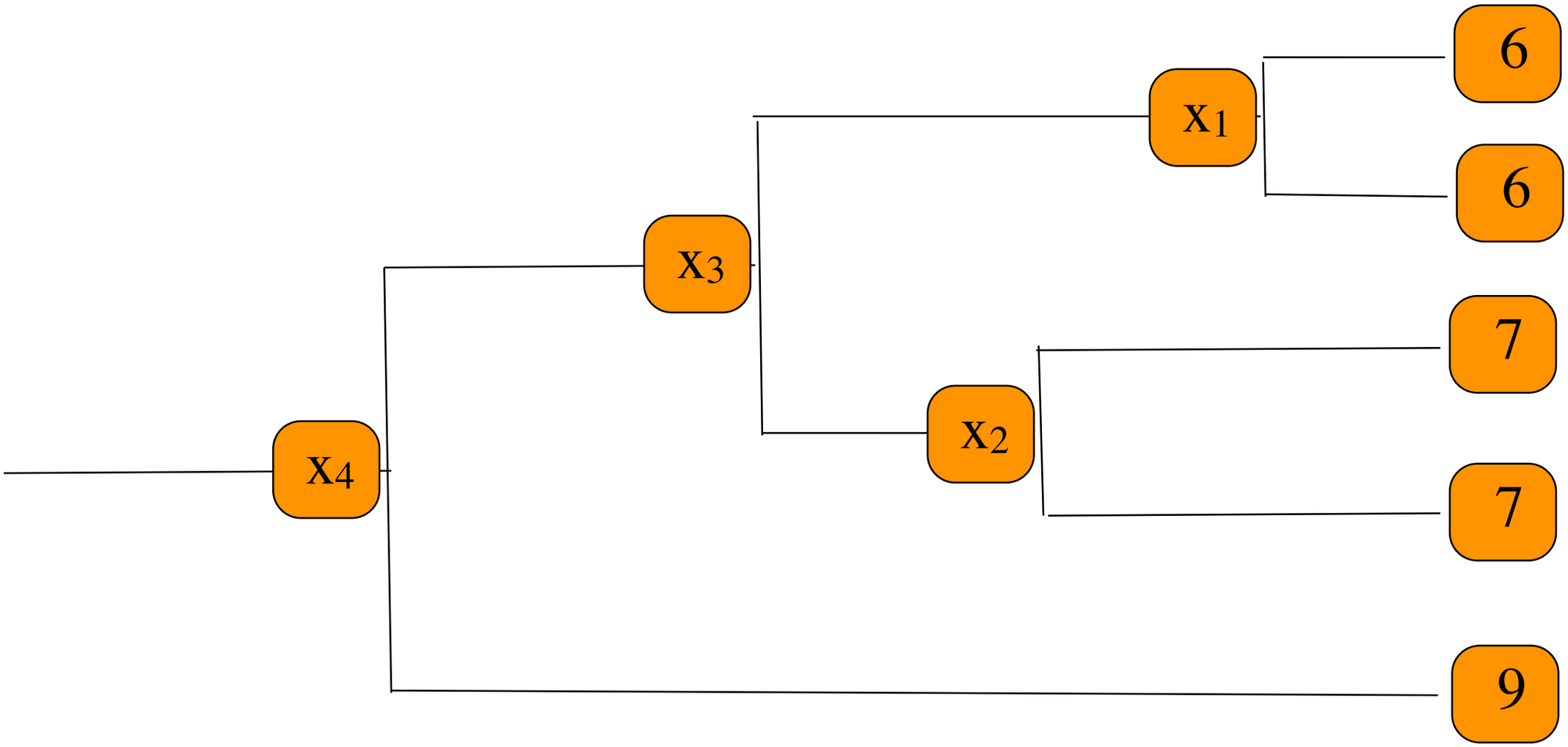
- Now, calculation of P(D|θ,Y) is possible, but is a difficult (i.e., very time-consuming) calculation ['Peeling' algorithm of Felsenstein]

- So, further augment tree Y to also include the state of every node....

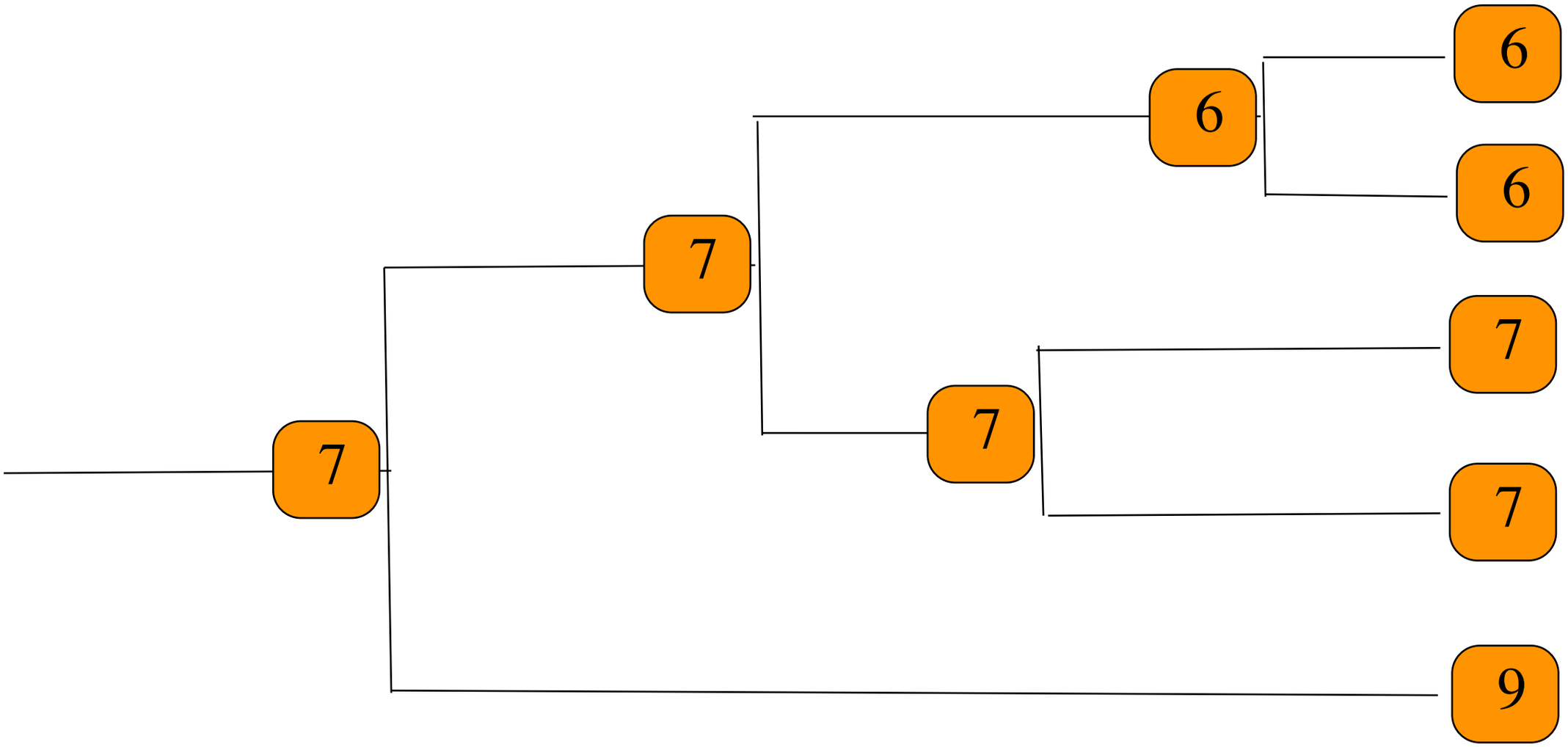
Augmented-tree



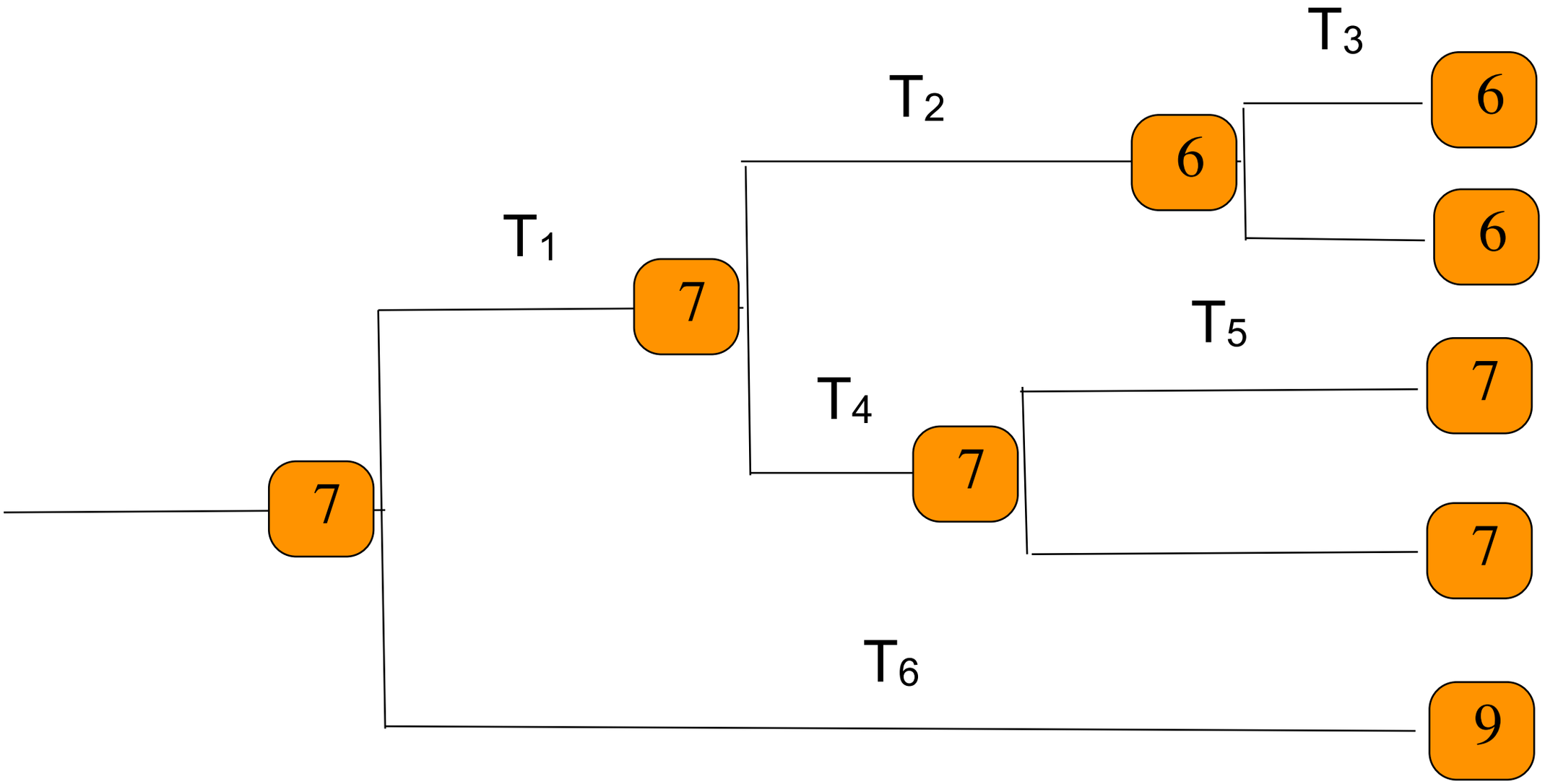
Augmented-tree



Augmented-tree



Augmented-tree



$P(Data|Tree) = P(MRCA=7)P(7->7|T_1)P(7->6|T_2)$
$P(6->6|T_3)^2\, P(7->7|T_4)P(7->7|T_5)^2\, P(7->9|T_6)$

- It remains to calculate P(i -> j | t)...
- Mutations are assumed to be symmetric [P(gain a repeat)=P(lose a repeat)]
- There are an infinite number of possible paths, each of which involves j-i directed mutations (assuming j>i) and d pairs of opposite mutations

e.g. for 5->7 we have paths such as

5->6->7

5->6->5->6->7

5->4->5->6->7->8->7

Equation 3

- Notation: t=time, θ/2= mutation rate
- (#mutns | t) ~ Poisson(tθ/2=λ)
- P(*m* mutns)= $\frac{\lambda^m e^{-\lambda}}{m!} = \frac{(\theta t/2)^m e^{-\theta t/2}}{m!}$
- P(i -> j | t,θ) = P(|i-j| |t,θ) [they assume symmetry]
- So, define d = |i-j|

Equation 3

$$P(d \mid t) = \sum_{k=0}^{\infty} P(2k+d \text{ mutns}|\theta, t)$$
$$\times P(k+d \text{ go 'up'}; k \text{ go 'down'})(\#\text{choices for the } k \text{ 'downs'})$$

$$= \sum_{k=0}^{\infty} \frac{e^{\frac{-\theta t}{2}} \left(\frac{\theta t}{2}\right)^{2k+d}}{(2k+d)!} \left[\left(\frac{1}{2}\right)^{k+d} \left(\frac{1}{2}\right)^{k}\right] \begin{pmatrix} 2k+d \\ k \end{pmatrix}$$

$$= e^{\frac{-\theta t}{2}} \sum_{k=0}^{\infty} \left(\frac{\theta t}{4}\right)^{2k+d} \frac{1}{(k+d)!d!}$$

Equation 3

$$P(d \mid t) = \sum_{k=0}^{\infty} P(2k+d \text{ mutns}|\theta, t) \times P(k+d \text{ go 'up'}; k \text{ go 'down'})(\#\text{choices for the } k \text{ 'downs'})$$

$$= \sum_{k=0}^{\infty} \frac{e^{\frac{-\theta t}{2}} \left(\frac{\theta t}{2}\right)^{2k+d}}{(2k+d)!} \left[\left(\frac{1}{2}\right)^{k+d} \left(\frac{1}{2}\right)^{k}\right] \binom{2k+d}{k}$$

$$= e^{\frac{-\theta t}{2}} \sum_{k=0}^{\infty} \left(\frac{\theta t}{4}\right)^{2k+d} \frac{1}{(k+d)!d!}$$

Claim: only first few terms matter

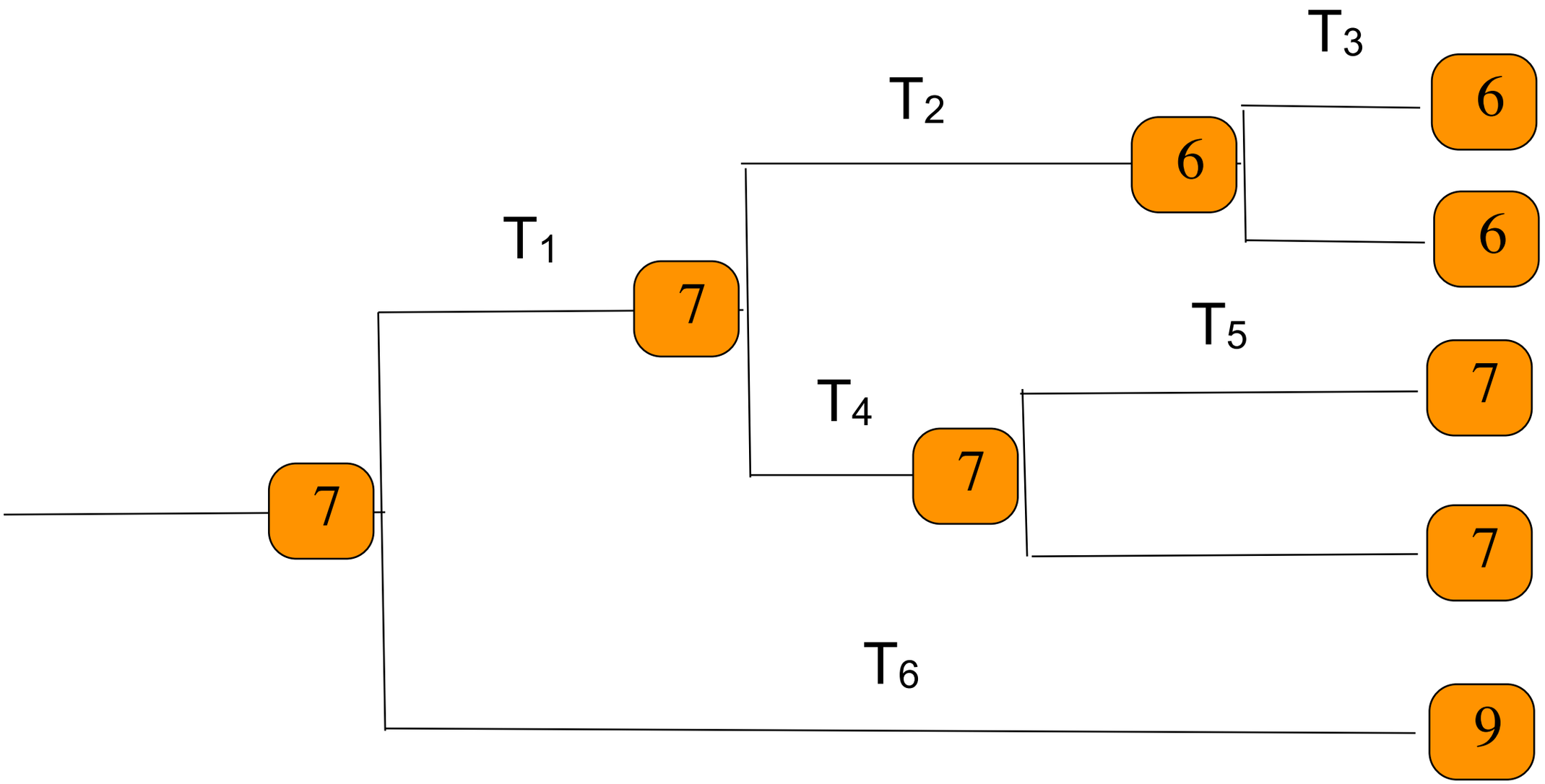
"modified d*th* order Bessel function of the first kind"

Proposal kernel

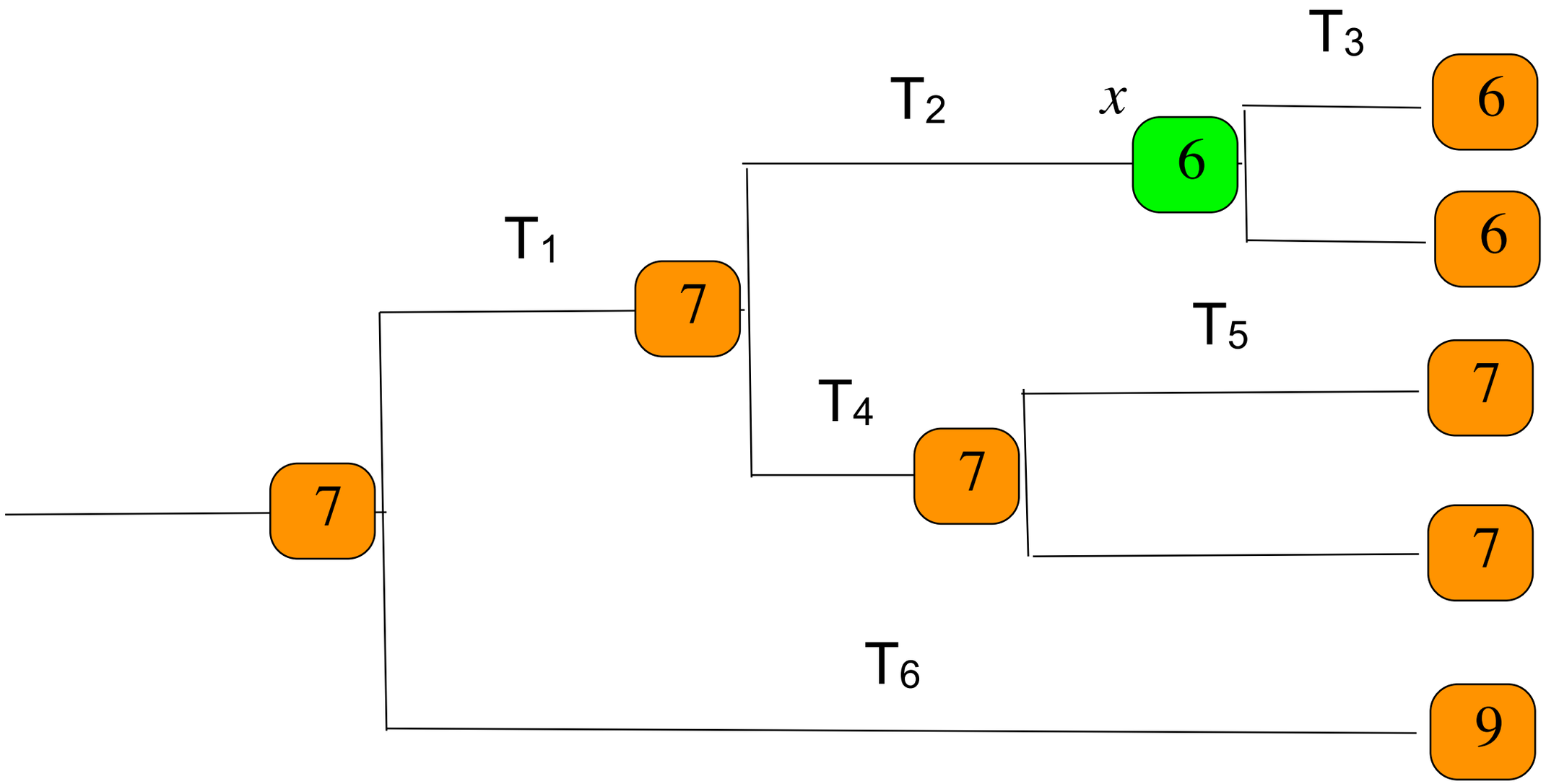
- State-space now consists of:

1. Tree topology
2. Times of events (i.e. nodes) on tree
3. State of ancestor at every node

Proposal kernel q, in pictures...

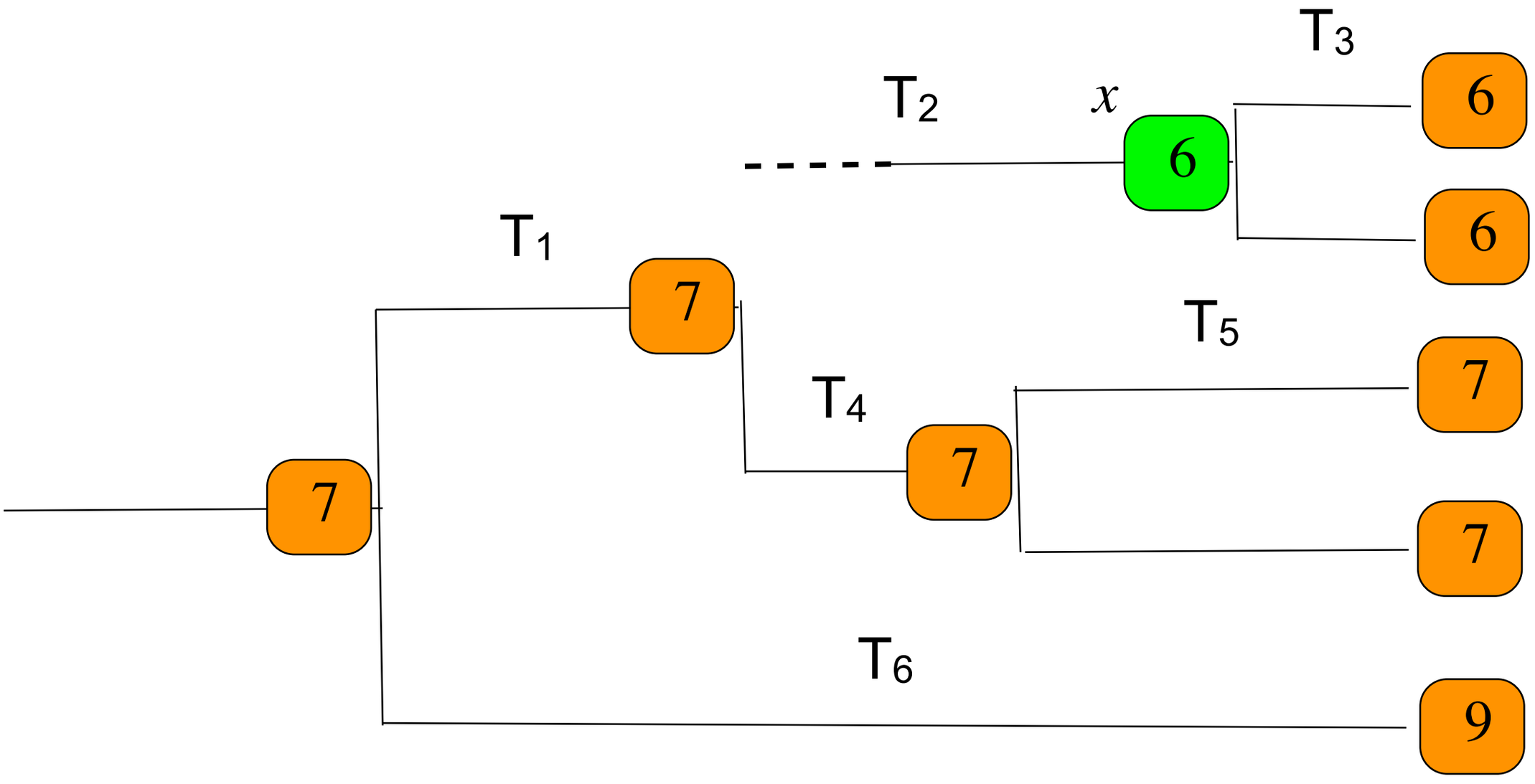


Proposal kernel q, in pictures...



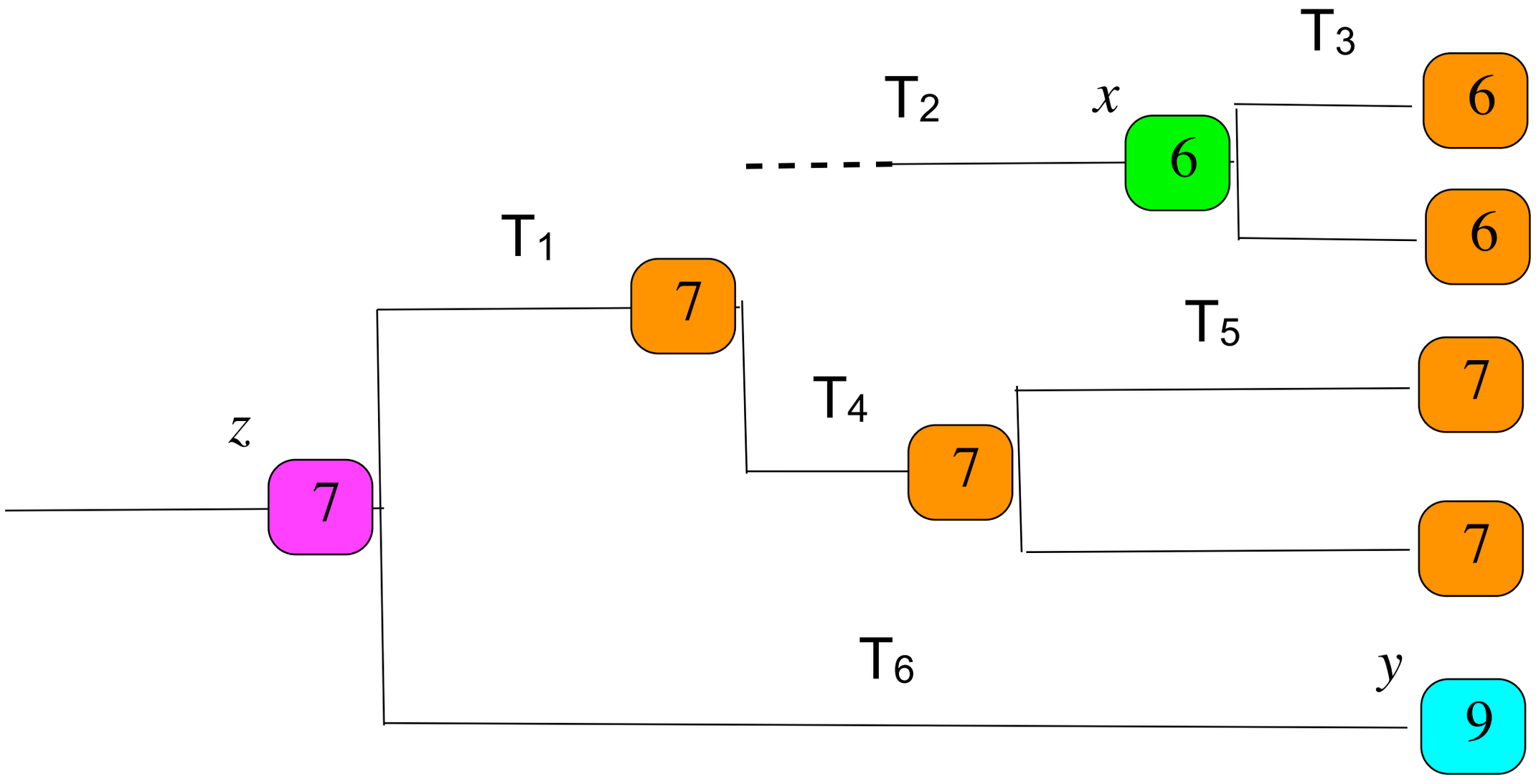
Pick an internal point (i.e. not a ‘leaf’) at random (call it *x*)

Proposal kernel q, in pictures...



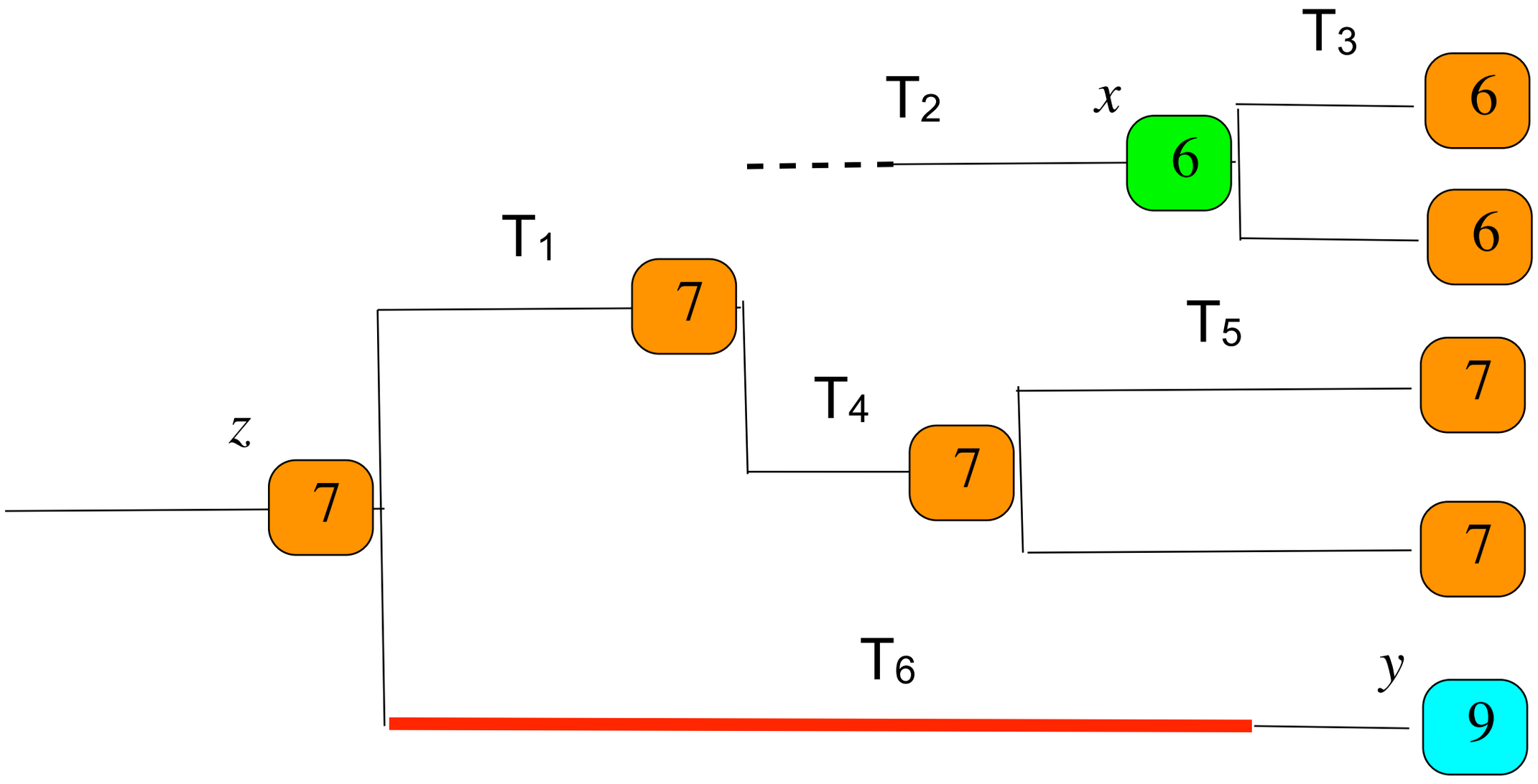
Detach *x* and its subtree

Proposal kernel q, in pictures...



Pick another random node, y, from the remaining portion of the tree, such that its parent, z, is higher up the tree than *x (i.e. t(z)>t(x))*. [*y* can be the root]

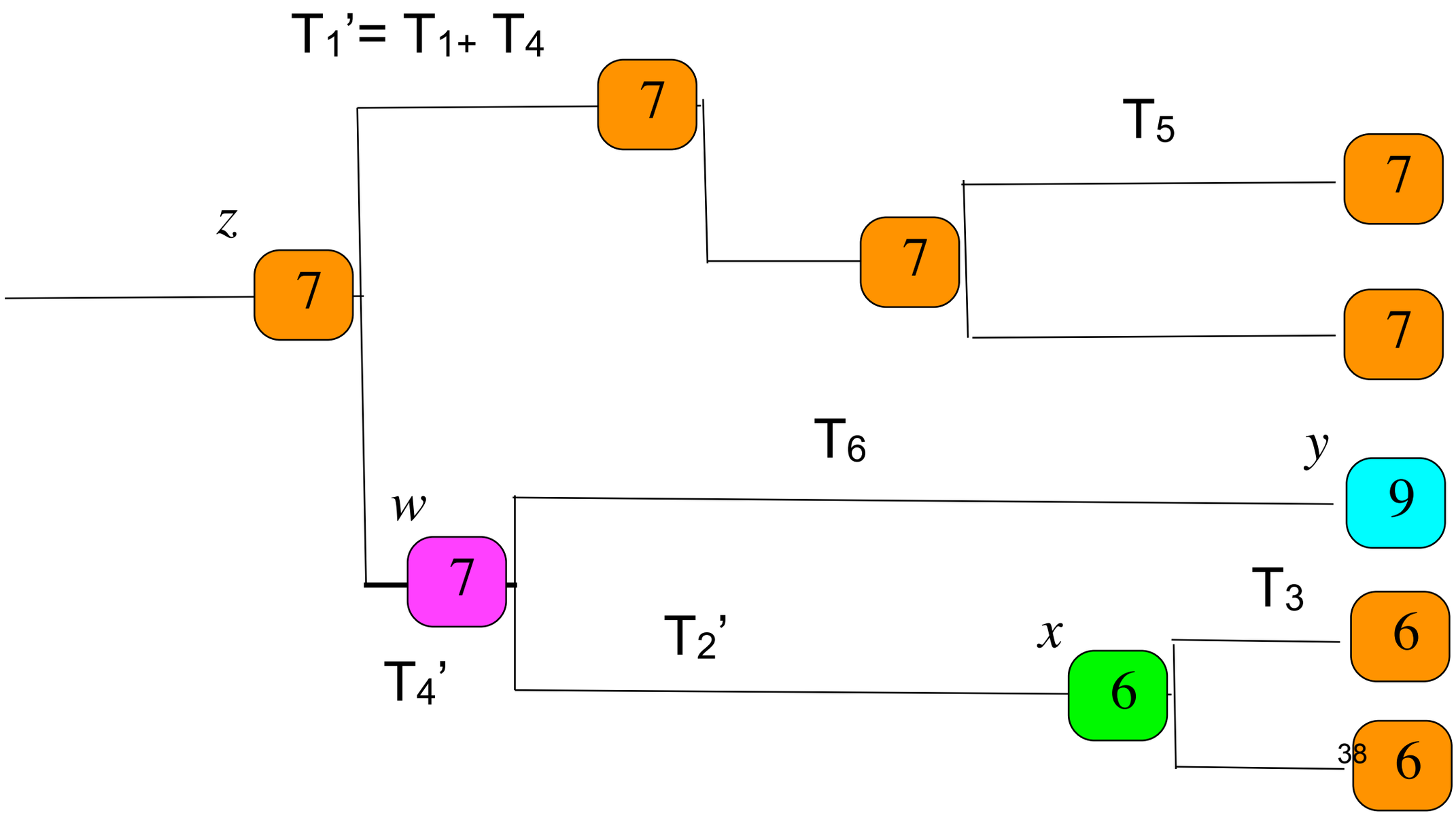
Proposal kernel q, in pictures...



Reattach x somewhere between $\max\{t(x),t(y)\}$ and $t(z)$ [Choose uniformly]

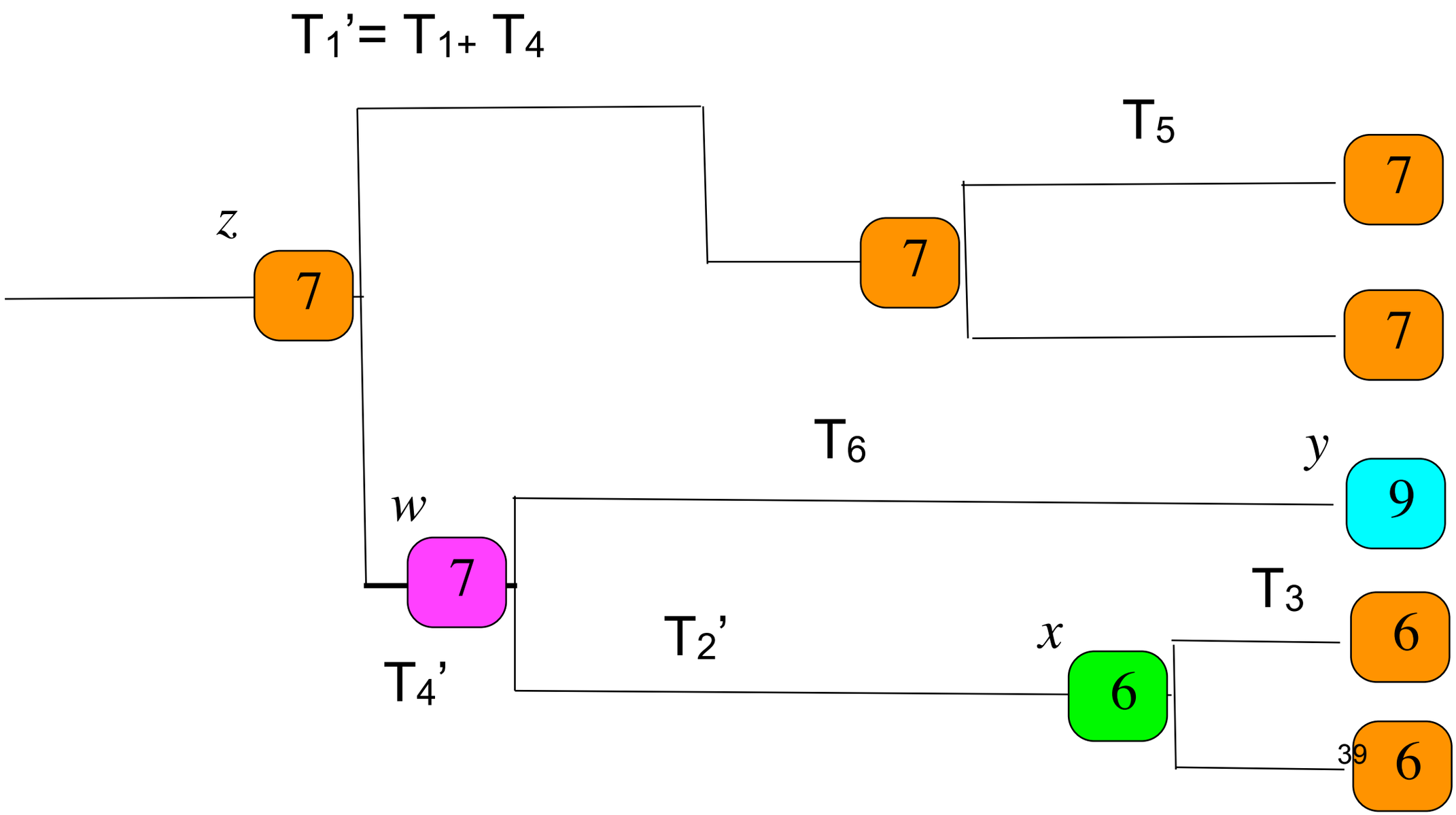
Proposal kernel q, in pictures...

Choose a state for the new node...(remove old node)



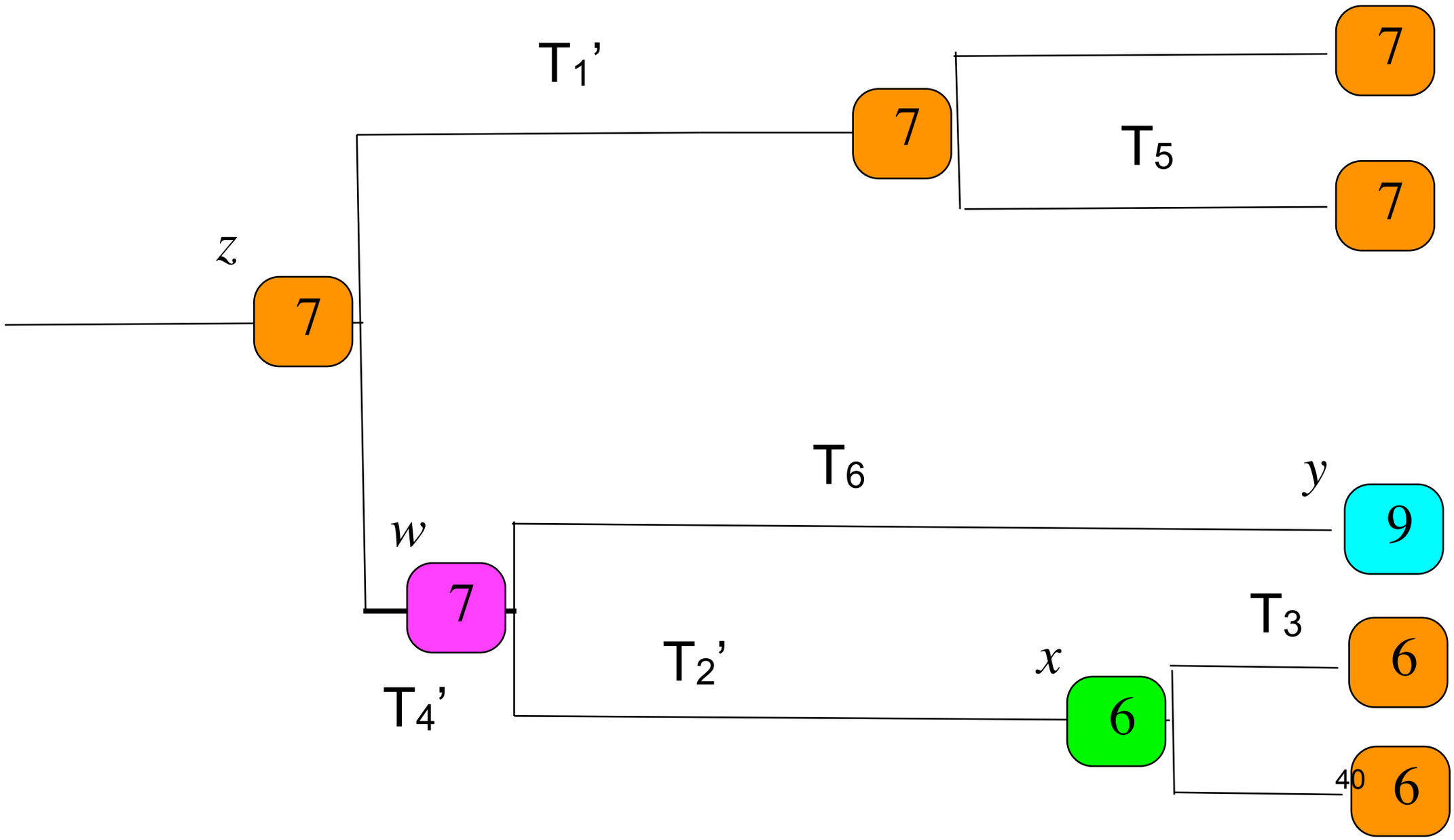
Proposal kernel q, in pictures...

Choose a state for the new node...(remove old node)



Proposal kernel q, in pictures...

Choose a state for the new node...(remove old node)



- Choice of node to attach above (*y*) is not uniform:

$$\mathrm{P}(\mathrm{y} \mid x) \propto \frac{1}{1+|\alpha(x)-\alpha(y)|}$$

where *α(x)* is the state of node x (etc.)

- Choice of state for the new node (*w*):

state~Discretized Normal with

mean= [*α(x)*+*α(y)*]/2,

std. dev.= (|*α(x)*-*α(y)*|+1)/4

- These choices are arbitrary **and are likely not the first things they tried!** Coming up with a good proposal kernel takes time. (Bad kernels mix poorly - i.e., the chain stays in the same state for a long time.)

- In order to know that the process will converge to f(θ|D) must have:
  - **Irreducible**: all trees can be reached by a sequence of such changes starting from any other tree
  - **Aperiodic**: There is no 'period' (trees can be reached at any iteration i, for i>j (some j)

- These are true for their choice of proposal kernel.

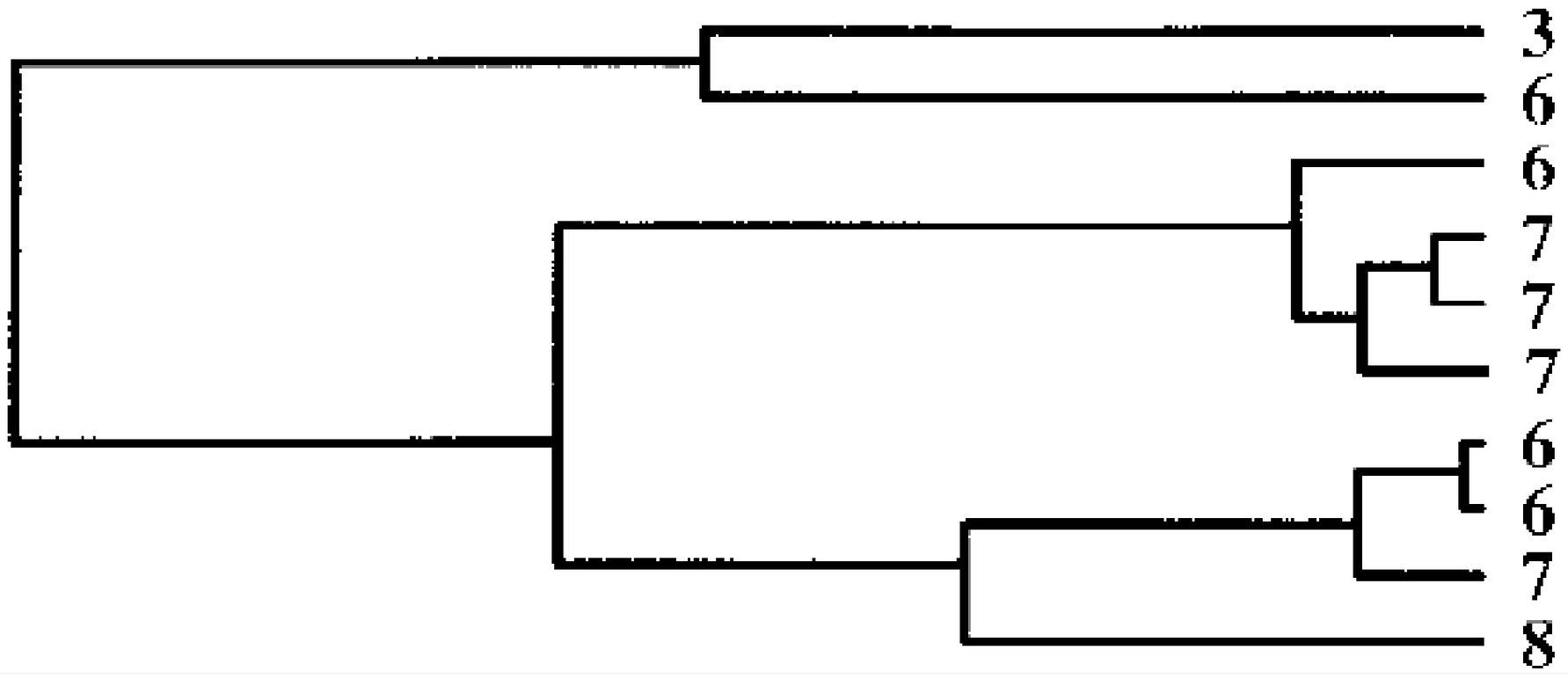
Actual implementation

Alternate two proposal steps:

1. Tree topology and state changes: as previously described.
2. Change θ (the mutation parameter.)

Analysis of simulated data (to benchmark performance)

Example (fig 1)



Tree simulated from their model (coalescent + step-wise mutation).

Note: Ancestry of the 6's (reconstruction will be challenging!)

$\theta=5$

Height =1.25 coalescent units (prior median =1.54);

Length = 4.82 (prior median =5.21)

Sample iterations:

Not much resemblance to true tree! Why not?

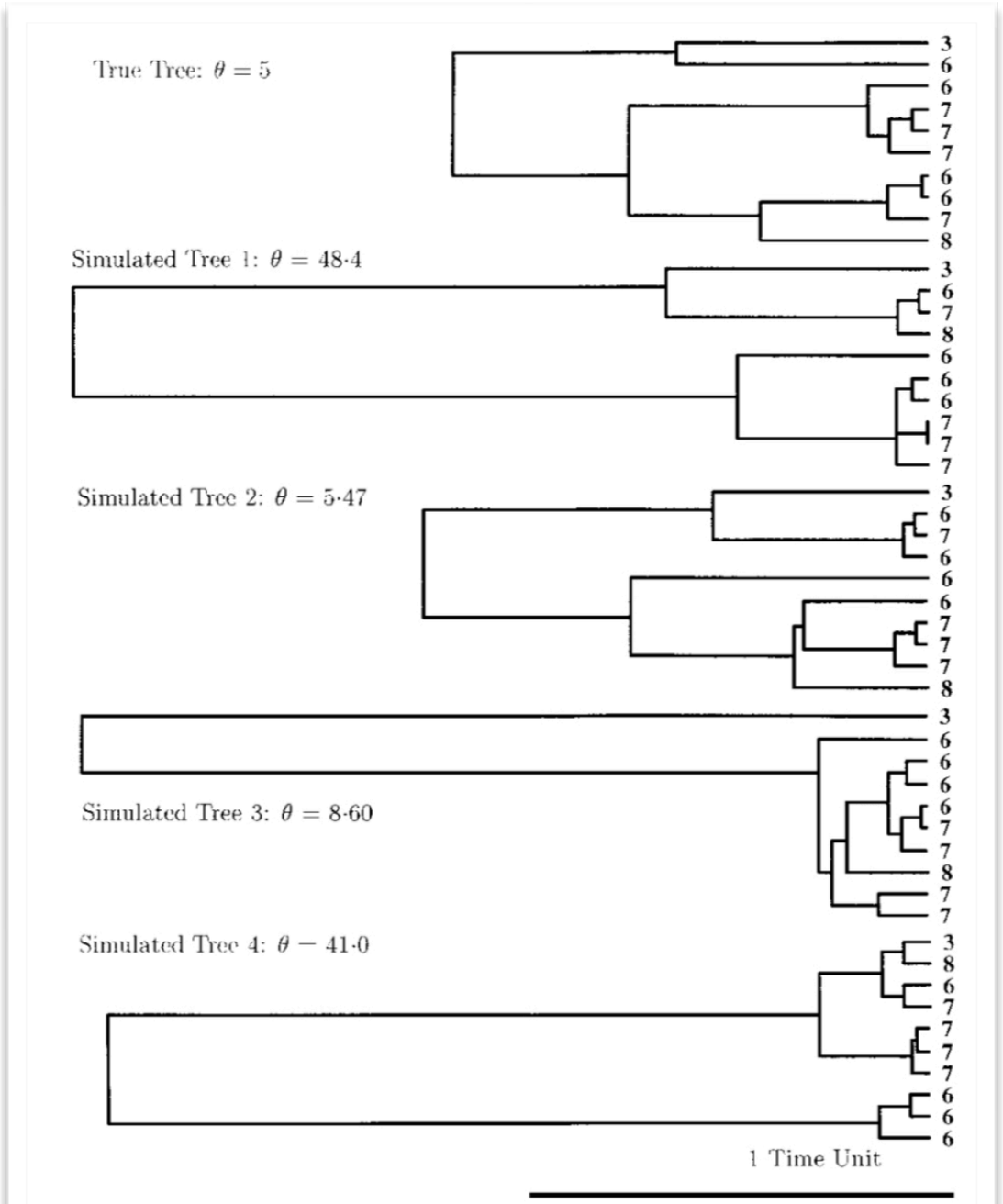


Figure 1.—The top tree ("true") is simulated from the coalescent-with-ladder-mutation model with $\theta = 5$. The other four trees are simulated from the postdata distribution given the allelic data of the true tree. These trees are samples numbered 2000, 4000, 6000, and 8000 from the MCMC run corresponding to row 1 of Table 1.

Results:

TABLE 1

Inferences for θ, T, and L from a single tree

		θ		T		L	
Sample size (n)	No. loci	Median	Interval	Median	Interval	Median	Interval
10	1	14.9	(2.9, 95)	1.32	(0.42, 4.0)	4.51	(1.8, 10)
40	1	13.0	(3.8, 38)	1.42	(0.55, 4.0)	7.33	(4.4, 13)
10	5	5.36	(2.3, 14)	1.33	(0.49, 3.3)	4.64	(2.0, 9)
40	5	5.67	(3.5, 9)	1.19	(0.58, 2.7)	6.59	(4.3, 11)

Median and 95% equal-tailed intervals of the posterior distributions for $\theta = 2N\mu$, tree height T, and total branch length L, based on samples of size $n = 10$, shown at the terminal nodes of the true tree of Figure 2, and $n = 40$ (not shown). The values of T and L are given in coalescent units; to obtain years, multiply by population size and generation time. The values used to generate the data were: $\theta = 5$, $T = 1.25$, $L = 4.82$ ($n = 10$), and $L = 7.15$ ($n = 40$). Table entries are estimated from 10,000 output values (corresponding to 2×10^5 attempts to update N and μ and 1.6×10^7 branch-swapping steps); simulation error is ~1–3% of stated values.

Row 1: Analysis of single tree of size 10

All posteriors have large variance

Results:

TABLE 1

Inferences for θ, T, and L from a single tree

		θ		T		L	
Sample size (n)	No. loci	Median	Interval	Median	Interval	Median	Interval
10	1	14.9	(2.9, 95)	1.32	(0.42, 4.0)	4.51	(1.8, 10)
40	1	13.0	(3.8, 38)	1.42	(0.55, 4.0)	7.33	(4.4, 13)
10	5	5.36	(2.3, 14)	1.33	(0.49, 3.3)	4.64	(2.0, 9)
40	5	5.67	(3.5, 9)	1.19	(0.58, 2.7)	6.59	(4.3, 11)

Median and 95% equal-tailed intervals of the posterior distributions for $\theta = 2N\mu$, tree height T, and total branch length L, based on samples of size $n = 10$, shown at the terminal nodes of the true tree of Figure 2, and $n = 40$ (not shown). The values of T and L are given in coalescent units; to obtain years, multiply by population size and generation time. The values used to generate the data were: $\theta = 5$, $T = 1.25$, $L = 4.82$ ($n = 10$), and $L = 7.15$ ($n = 40$). Table entries are estimated from 10,000 output values (corresponding to 2×10^5 attempts to update N and μ and 1.6×10^7 branch-swapping steps); simulation error is ~1–3% of stated values.

Row 2: Tree of 40 indivs.

Variance for θ reduced, other variances largely unchanged (Tree height T was actually the same)

5 completely linked micro-satellite loci:

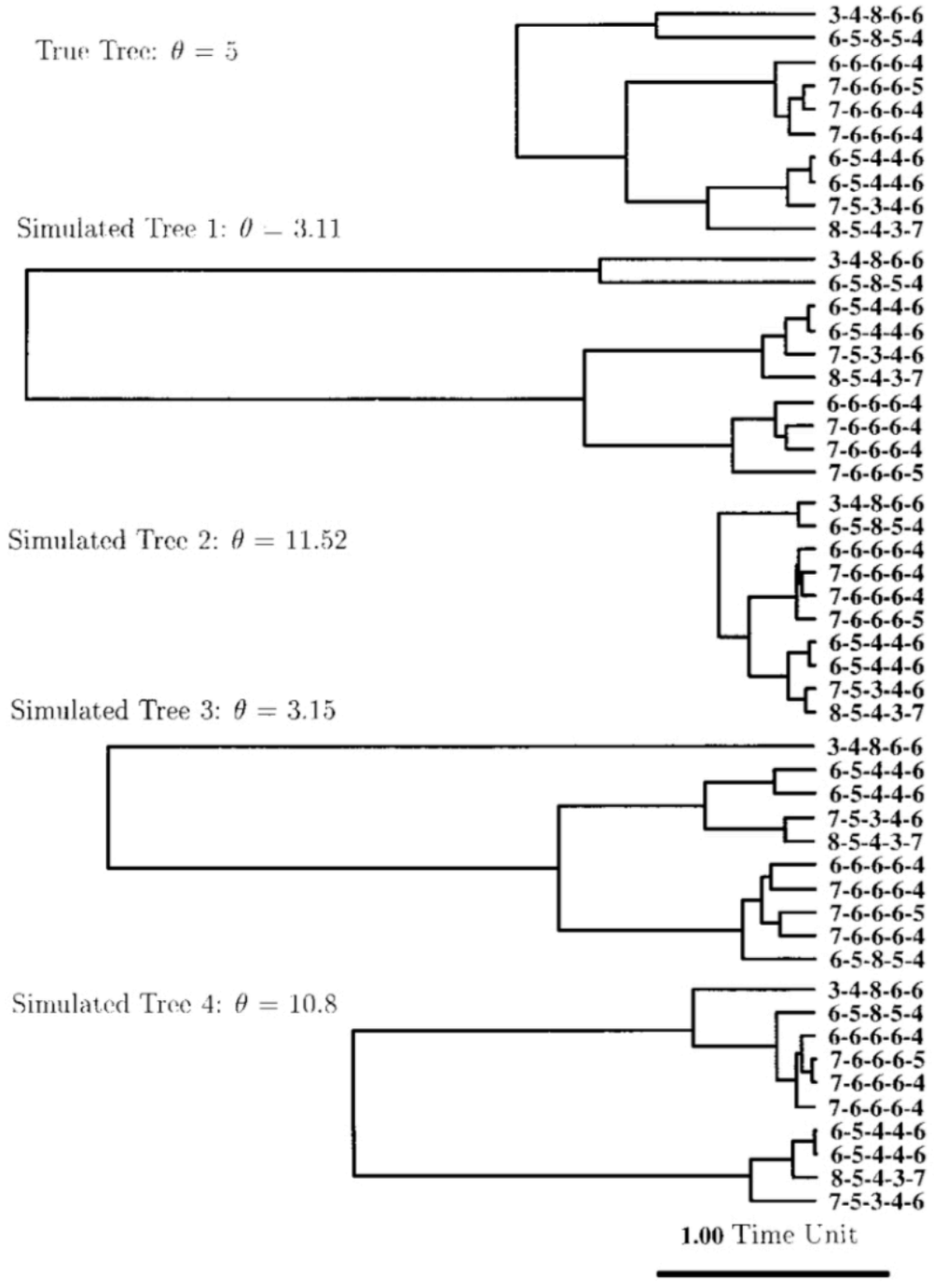


Figure 2.—The true tree (top) is the same as that of Figure 1, but the results of four additional, independent simulations of the mutation process are also shown, mimicking data from five completely linked loci, each having the same mutation mechanism and with $\theta = 5$. The other four trees are simulated from the postdata distribution given all five data sets. These trees are samples numbered 2000, 4000, 6000, and 8000 from the MCMC run corresponding to row 3 of Table 1.

Results:

TABLE 1

Inferences for θ, T, and L from a single tree

		θ		T		L	
Sample size (n)	No. loci	Median	Interval	Median	Interval	Median	Interval
10	1	14.9	(2.9, 95)	1.32	(0.42, 4.0)	4.51	(1.8, 10)
40	1	13.0	(3.8, 38)	1.42	(0.55, 4.0)	7.33	(4.4, 13)
10	5	5.36	(2.3, 14)	1.33	(0.49, 3.3)	4.64	(2.0, 9)
40	5	5.67	(3.5, 9)	1.19	(0.58, 2.7)	6.59	(4.3, 11)

Median and 95% equal-tailed intervals of the posterior distributions for $\theta = 2N\mu$, tree height T, and total branch length L, based on samples of size $n = 10$, shown at the terminal nodes of the true tree of Figure 2, and $n = 40$ (not shown). The values of T and L are given in coalescent units; to obtain years, multiply by population size and generation time. The values used to generate the data were: $\theta = 5$, $T = 1.25$, $L = 4.82$ ($n = 10$), and $L = 7.15$ ($n = 40$). Table entries are estimated from 10,000 output values (corresponding to 2×10^5 attempts to update N and μ and 1.6×10^7 branch-swapping steps); simulation error is ~1–3% of stated values.

Row 3: 5 linked loci, sample size=10

Results:

TABLE 1

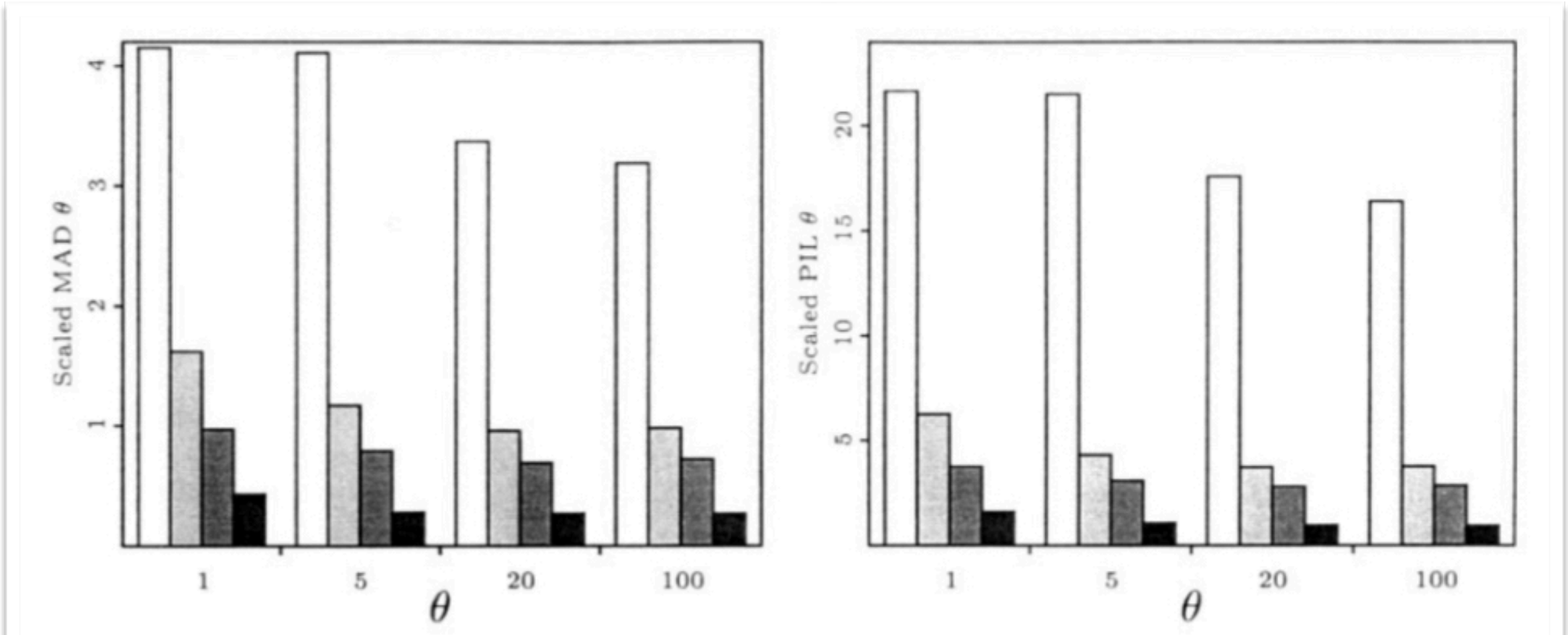
Inferences for θ, *T*, and *L* from a single tree

		θ		*T*		*L*	
Sample size (*n*)	No. loci	Median	Interval	Median	Interval	Median	Interval
10	1	14.9	(2.9, 95)	1.32	(0.42, 4.0)	4.51	(1.8, 10)
40	1	13.0	(3.8, 38)	1.42	(0.55, 4.0)	7.33	(4.4, 13)
10	5	5.36	(2.3, 14)	1.33	(0.49, 3.3)	4.64	(2.0, 9)
40	5	5.67	(3.5, 9)	1.19	(0.58, 2.7)	6.59	(4.3, 11)

Median and 95% equal-tailed intervals of the posterior distributions for $\theta = 2N\mu$, tree height T, and total branch length L, based on samples of size $n = 10$, shown at the terminal nodes of the true tree of Figure 2, and $n = 40$ (not shown). The values of T and L are given in coalescent units; to obtain years, multiply by population size and generation time. The values used to generate the data were: $\theta = 5$, $T = 1.25$, $L = 4.82$ ($n = 10$), and $L = 7.15$ ($n = 40$). Table entries are estimated from 10,000 output values (corresponding to 2×10^5 attempts to update N and μ and 1.6×10^7 branch-swapping steps); simulation error is ~1–3% of stated values.

Row 4: 5 linked loci, sample size=40

Figure 3: results averaged over 140 data sets for θ:

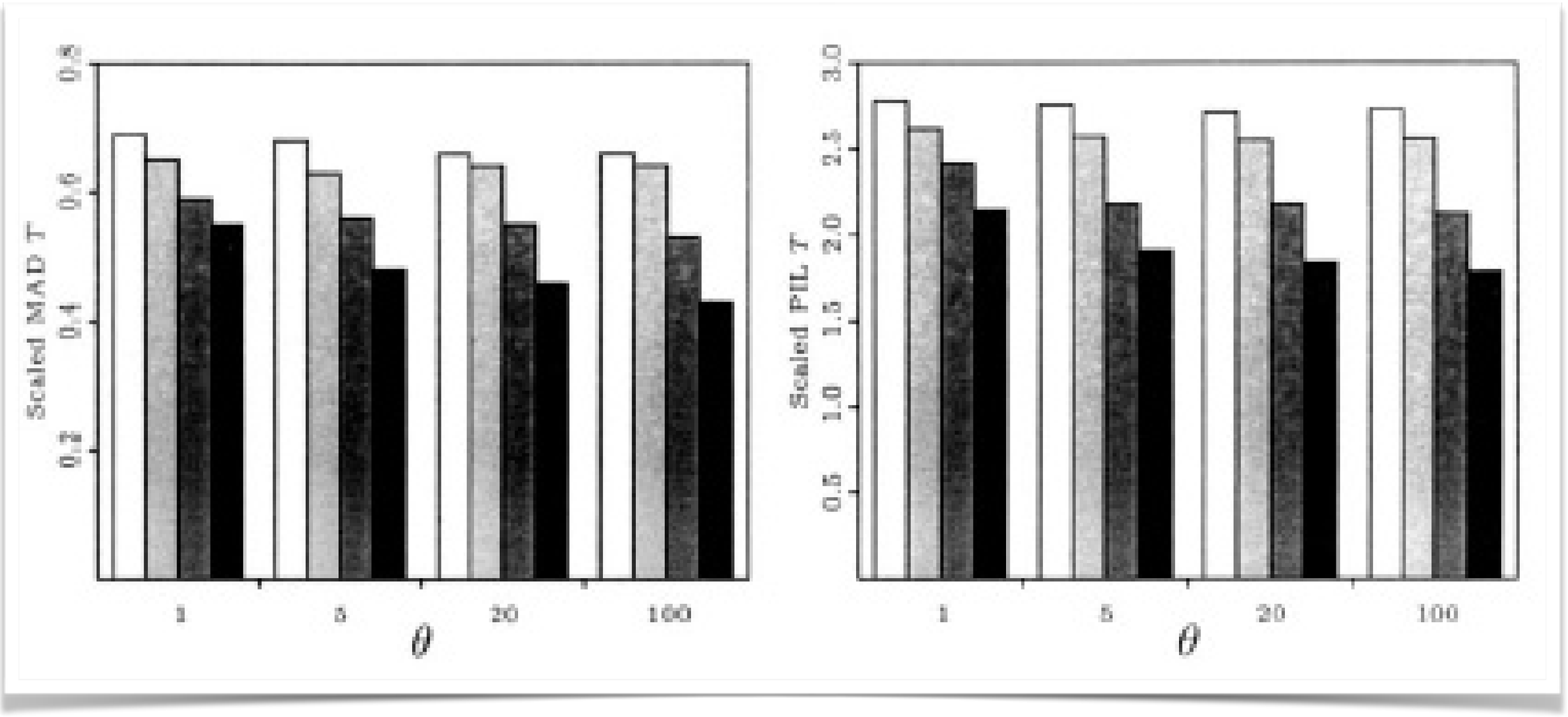


Mean absolute difference [MAD] and 95% probability interval length [PIL] for θ (scaled by true θ)

Bar 1 = single locus, n=10; Bar 2 = single locus, n=40

Bar 3 = 5 loci, n=10; Bar 4 = 5 loci, n=40.

Averaged over 140 data sets: for *T*



Mean absolute difference [MAD] and 95% probability interval length [PIL] for T (scaled by true T)

Bar 1 = single locus, n=10; Bar 2 = single locus, n=40

Bar 3 = 5 loci, n=10; Bar 4 = 5 loci, n=40.

Actual application: Y-chromosome Adam

- Data of Cooper *et al.* (1996)
- 212 individuals (E. Anglia, Sardinia, Nigeria)
- 5 micro-satellite loci
- Two analyses:
  1. Roughly 20 indivs from each population [NSE]
  2. All 174 E. Anglians (only) [EA]

Prior distributions

- θ=2Nμ, where μ is the actual mutation rate and N is the (effective) population size. (confounding)
- Heyer *et al.* (1997) observed 3 mutations in 1491 meioses --> μ~0.02. Used a gamma prior with mode 3/1492 and mean 4/1492
- Two priors for N, both centered at ~5000:
  "low -variance" - Gamma (concentrated on [1000,10000])
  "high variance" - Lognormal (some support for values >20000)
- Coalescent prior for *T*.

MCMC details

- 40 branch swaps between each attempt to update *N* and *μ*
- **Sample once every 4100 iterations to avoid correlation(!)**
- Discard first 2000 such samples ('burn-in'), retain next 10000 such samples
- Do two analyses for each scenario (starting from a different, randomly generated tree) [Stationarity test]

TABLE 2

Summary of human *Y* chromosome analyses

		Low-variance prior		High-variance prior	
		Median	Interval	Median	Interval
θ	Prior	22.0	(4.8, 75.9)	39.2	(4.0, 338)
	NSE	11.4	(7.7, 17.0)	11.2	(7.6, 16.4)
	EA	10.0	(7.4, 13.3)	9.8	(7.3, 13.1)
μ	Prior	2.5	(0.73, 5.9)	2.5	(0.73, 5.9)
($\times 10^{-3}$)	NSE	1.7	(0.74, 3.7)	1.8	(0.59, 4.6)
	EA	1.5	(0.67, 3.5)	1.8	(0.57, 4.3)
N	Prior	4.7	(1.6, 10.3)	8.2	(1.1, 56.4)
($\times 10^{3}$)	NSE	3.5	(1.5, 7.4)	3.0	(1.1, 9.6)
	EA	3.3	(1.4, 7.1)	2.7	(1.1, 8.6)
TMRCA	Prior ($n = 60$)	157	(39, 579)	281	(31, 2466)
($\times 10^{3}$ yr)	NSE	36	(13, 128)	33	(10, 138)
	Prior ($n = 174$)	159	(39, 565)	289	(32, 2493)
	EA	31	(11, 108)	27	(8.7, 113)

Median and 95% equal-tailed intervals of prior and posterior distributions for θ, μ *N*, and TMRCA for the NSE sample (60 *Y* chromosome haploptyes, approximately equal numbers from Nigeria, Sardinia, and East Anglia), and for the EA sample (174 East Anglian haplotypes). Haplotypes consist of five microsatellite loci; data from Cooper *et al.* (1996). Prior distributions are: $\mu \sim$ gamma (4,1492); $N \sim$ gamma (5,1/ 1000) (low variance), and $N \sim$ ln (9,1) (high variance). Table entries are based on 10,000 output values (corresponding to 4×10^7 branch-swapping steps).

Robustness to choice of prior

In summary

- They built a full probability model to allow exact calculation of P(D|θ) (subject to correctness of the model itself).
- Augmented state-space to make calculation possible. *This is often a useful tool.*
- Multiple (tightly linked) loci are required for useful inference.
- Inference for the two datasets was quite similar despite geographical differences.
- TMRCA estimates substantially lower than earlier studies from mtDNA, but broadly consistent with Tavaré *et al.* (1997 - Y chromosome) - stochastic variation?

Another (Simpler!) 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.

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

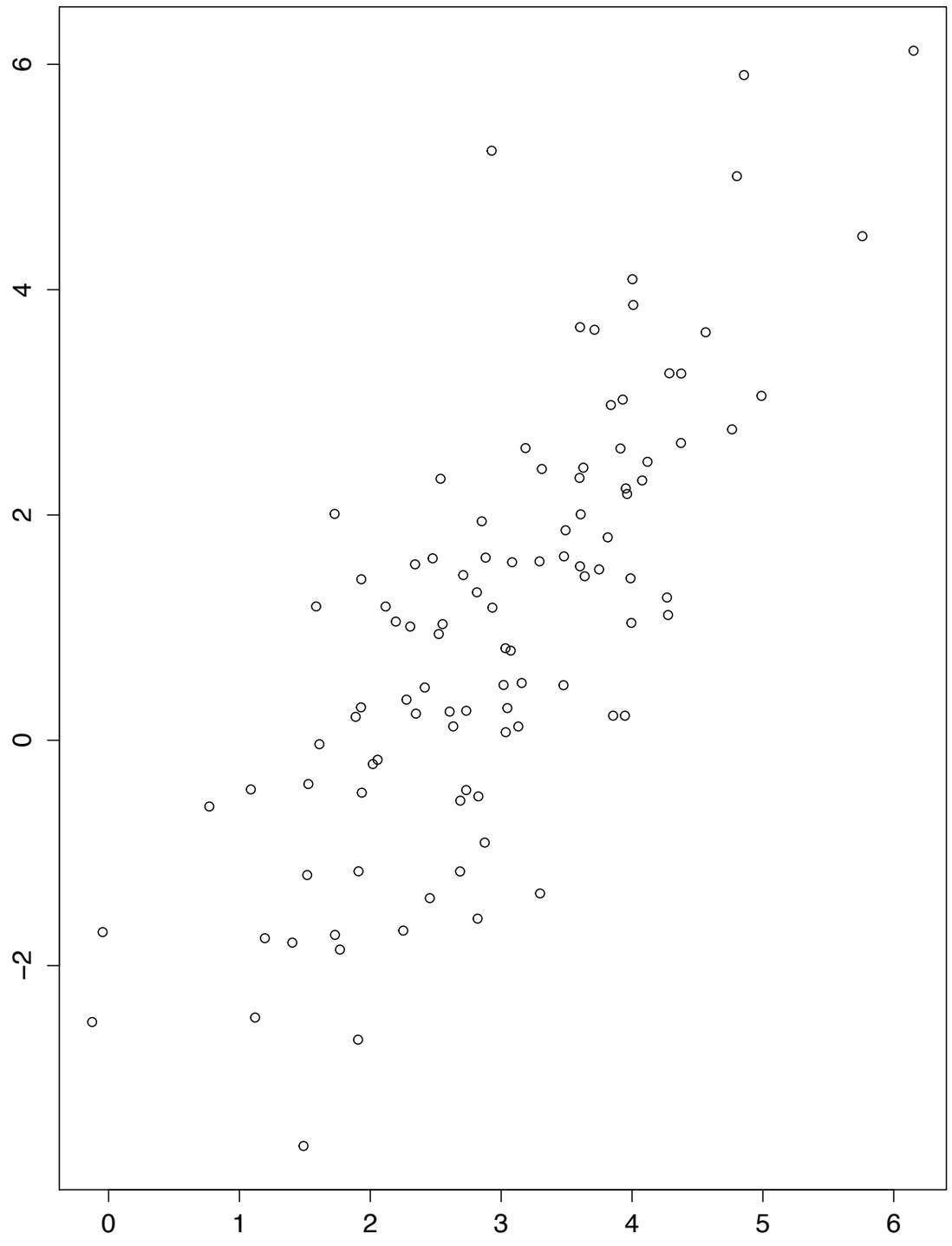
Properties:

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

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

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

The data (correlation=0.75)



MVNormals.Rmd on Github

```
# Generate some test data
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))
# Now generate one hundred samples from that distribution:
our.data<-rmvnorm(n=100,mean=mu.vector,sigma=variance.matrix)
```

```
# how many iterations do we want in our MH-MCMC process?
total.iterations<-10000

do.MHMCMC<-function(number.of.iterations){
  # start our MH-MCMC process off from somewhere
  current.mu<-c(3,1)
  current.mu<-runif(2,0,4)

  # define a vector to store the output of the MH-MCMC process….
  posterior.mu<-mat.or.vec(number.of.iterations,2)
```

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.

MCMC in R

- Metrop library.
- See Week7—MCMCPackage repo on Github.

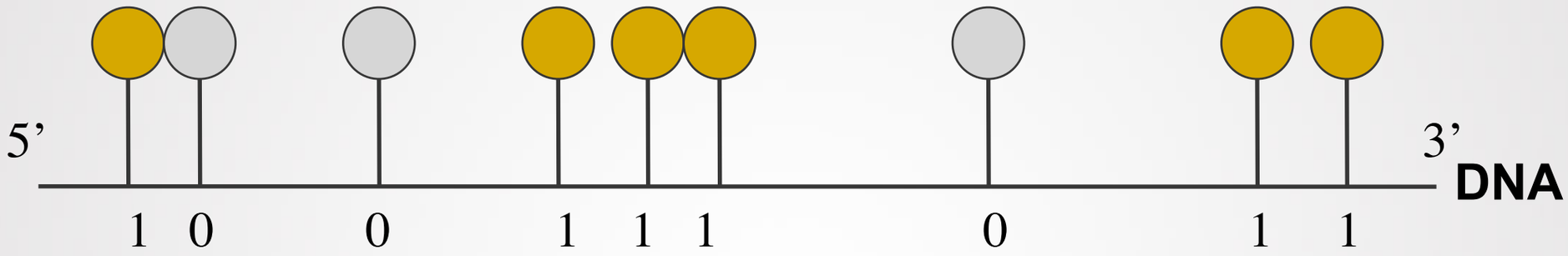
Lab time

Extra example MCMC application:

R21: Conservation and functional-characterization of tumor methylation sites

Investigators: Marjoram, Siegmund, Shibata

DNA methylation



= CpG

= mCpG

Methylation status of neighboring CpGs is correlated

Specific Aims

Specific Aim 1: Statistical methods for detecting conservation of DNA methylation status. We will develop statistical methods to classify CpG sites as stable or unstable from normal colon. **[Purely statistical approaches]**

Specific Aim 2: Evolutionary modeling of tumor evolution, with the goal of classification of CpG sites, genes and pathways as stable or unstable, and reconstructing the state of the first tumor cell. **[Model-based simulation and ABC analysis]**

Specific Aim 3: Develop statistical methods to classify genes and cellular pathways more conserved during growth and therefore more likely to be functional. **[i.e., extend Spec. Aim 1 to genes]**

Specific Aim 4: Software development and distribution. **[R-packages and Shiny apps]**.

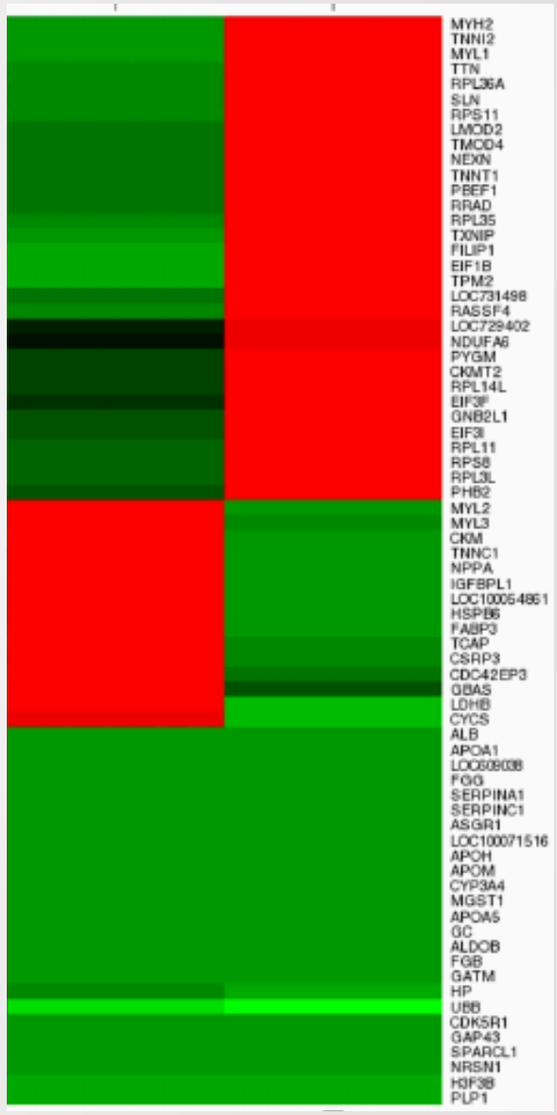
Data - Darryl Shibata

- Multi-regional sampling from each tumor - ‘left’ and ‘right’ side.
  - Often just 1 sample from the left and one from the right, but sometimes we will have more.
- For many tumors, we will have matched ‘normal’ data.
- Illumina Infinium MethylationEPIC BeadChip Kit: measures methylation at ~850,000 CpG sites.
- **Important caveat: our data is pooled, rather than cell-level.**
  - So, we will have sample-level methylation proportions at each site, rather than ‘yes’/‘no’ [1/0] cell-level data.

Methods for detecting differential methylation

- Goal: To detect differential patterns of methylation
  - e.g., tumor vs. normal

- A variety of methods exist:
  - Focus is on differences in mean expression
  - Most ignore correlation between neighboring CpGs, until…



tumor normal

Jenkinson *et al. BMC Bioinformatics* (2018) 19:87
https://doi.org/10.1186/s12859-018-2086-5

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

An information-theoretic approach to the modeling and analysis of whole-genome bisulfite sequencing data

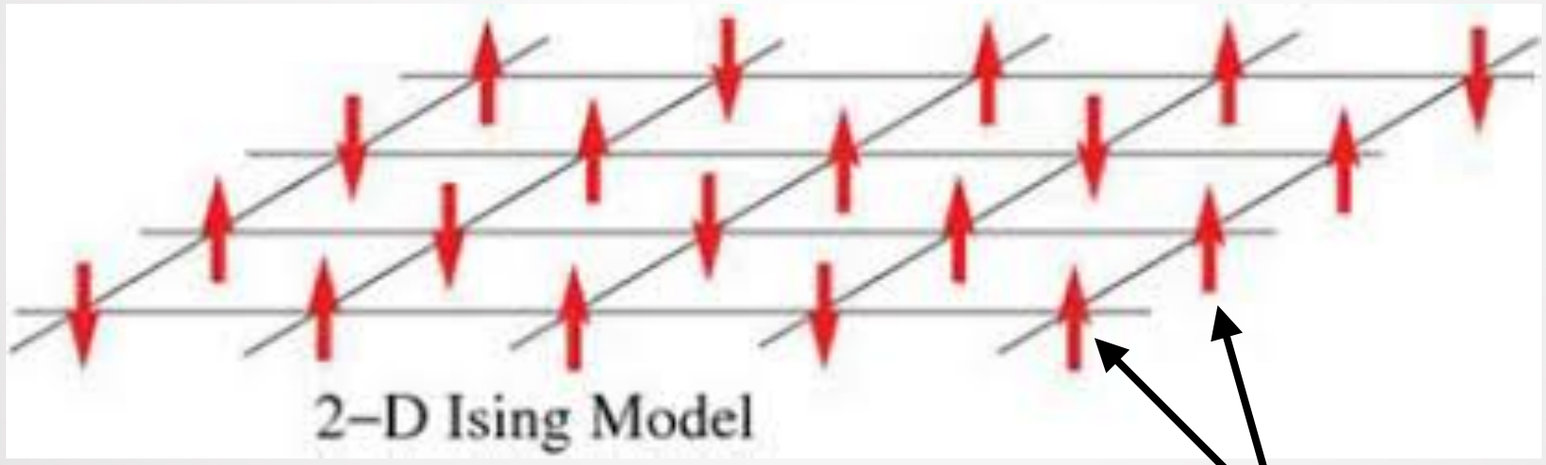
Garrett Jenkinson[1,2], Jordi Abante[1], Andrew P. Feinberg[2,3,4] and John Goutsias[1*]

Abstract

Background: DNA methylation is a stable form of epigenetic memory used by cells to control gene expression. Whole genome bisulfite sequencing (WGBS) has emerged as a gold-standard experimental technique for studying DNA methylation by producing high resolution genome-wide methylation profiles. Statistical modeling and analysis is employed to computationally extract and quantify information from these profiles in an effort to identify regions of

Ising Model

- Model of ferro-magnetism
- Lattice of atoms, each of which can be pointing “up” (σ_i = +1) or “down” (σ_i = -1)
- Two forces acting on lattice:

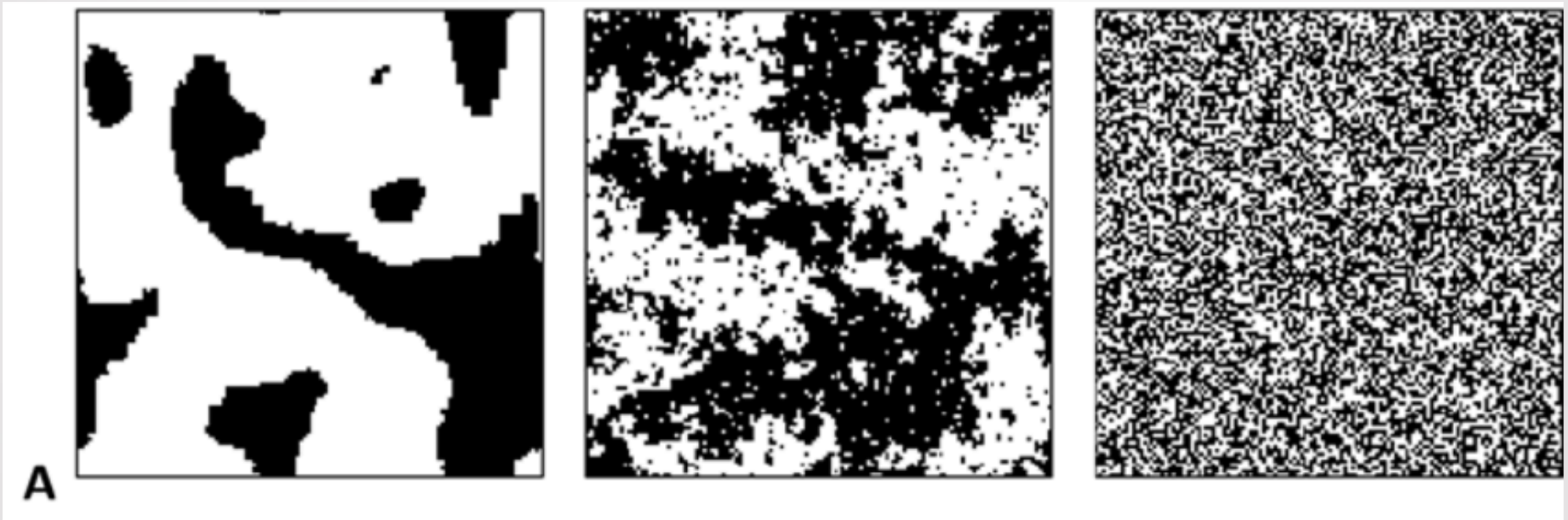


External force (acts globally)

Correlation between neighbors (acts locally)

Ising models in NetLogo

- NetLogo is an agent-based simulation package…



Ising Model

- Lattice of atoms, each of which can be pointing "up" (σ= +1) or "down" (σ= -1)
- Typically, define state probability (simplifying slightly) as:

$$P(\sigma) = exp[\mu \sum_{j} h_j \sigma_j \; + \; \sum_{<i,j>} J_{i,j} \; \sigma_i \; \sigma_j] / Z,$$

Sum over all adjacent pairs of atoms

- where $Z = \sum exp(H(\sigma))$ which is non-trivial to calculate.

Sum over all possible configurations of atoms

μ: global tendency to point "up"
h_j: atom-specific response to μ
J_{ij}: pairwise-correlation between atoms i & j

MCMC sampling from Ising models

- For large lattices, producing sample configurations is non-trivial. (Normalizing constant cannot be calculated.)
- Simple MCMC algorithm to produce sample configurations (i.e., σ) from an Ising model:
- Given a transition(proposal) kernel q(σ -> σ').
- The Hastings Ratio for a move from σ to σ' is

$$h = \min\left\{1, \frac{q(\sigma' \to \sigma)\ P(\sigma')}{q(\sigma \to \sigma')\ P(\sigma)}\right\}$$

$$= \min\left\{1, \frac{q(\sigma' \to \sigma)\ exp(H(\sigma'))/\cancel{Z}}{q(\sigma \to \sigma')\ exp(H(\sigma))/\cancel{Z}}\right\}$$

Jenkinson et al.

- Their analysis is based on Ising model, so explicitly captures correlation between neighboring loci.
- Their lattice is one-dimensional (i.e., CpGs along the genome).
- Breaks the genome into short regions (n=20-40 CpGs) - so the normalizing constant can be calculated (2^n terms in summation).
- Estimate Ising model parameters for each region (external force represents P(meth); local force reflects correlation).
- They smooth the estimates (noisy because regions are short).
- Look for regions (genes) in which parameter estimates differ between normal and tumor samples (say).

Part of our proposal…

- **Our data is pooled**, but let's use the Ising model "energy" formulation anyway.
- x=(x_1,x_2,…,x_n) is the methylation configuration in region k.
- Define configuration prob. as

$$P_X(\boldsymbol{x}) = \frac{1}{Z} \exp\left\{ \sum_{n=1}^{N} a_n(2x_n - 1) + \sum_{n=2}^{N} c_n(2x_n - 1)(2x_{n-1} - 1) \right\}, \quad (1)$$

- where

$$a_n = \alpha_k + \beta_k \rho_n, \quad (3)$$

and

$$c_n = \frac{\gamma_k}{d_n},$$

α_k: region-specific tendency to be methylated
ρ_n: density of CpGs at n^{th} CpG
β_k: region-specific dependence on density
d_n: distance between n^{th} CpG and $(n-1)^{th}$ CpG
γ_k: region-specific correlation parameter

First problem:

$$P_X(\boldsymbol{x}) = \frac{1}{Z} \exp\left\{ \sum_{n=1}^{N} a_n (2x_n - 1) + \sum_{n=2}^{N} c_n (2x_n - 1)(2x_{n-1} - 1) \right\}, \quad (1)$$

$$a_n = \alpha_k + \beta_k \rho_n,$$

and

$$c_n = \frac{\gamma_k}{d_n},$$

α_k: region-specific tendency to be methylated
ρ_n: density of CpGs at n^{th} CpG
β_k: region-specific dependence on density
d_n: distance between n^{th} CpG and $(n-1)^{th}$ CpG
γ_k: region-specific correlation parameter

- 3 parameters to estimate per region - and we only have 1 sample (or maybe 2).
- Solution: hierarchical model? So $\alpha_k \sim$ Normal(m,ν^2). Allow two populations/groups? (e.g., normal and conserved?)
  - Hierarchy across samples (for one region) or across regions (for one sample)?
  - Across samples will be much easier because of the normalizing constant.

Second problem:

- The normalizing constant:
  - Jenkinson: $$Z = \sum exp(H(\underline{x}))$$

n sites gives 2^n terms

  - Us: $$Z = \int \int \cdots \int exp(H(\underline{x}))d\underline{x}$$

n sites gives n-dimensional integral

Solution 1 (non-pooled setting):

- When finding posterior for parameters using MCMC, when proposing a move from parameter θ to θ' we need to evaluate

$$
\begin{aligned}
h &= \min\left\{1, \frac{q(\theta' \to \theta)\ P(\theta' \mid X)}{q(\theta \to \theta')\ P(\theta \mid X)}\right\} \\
&= \min\left\{1, \frac{q(\theta' \to \theta)\ \pi(\theta')\ exp(H_{\theta'}(X))/Z_{\theta'}}{q(\theta \to \theta')\ \pi(\theta)\ exp(H_{\theta}(X))/Z_{\theta}}\right\}
\end{aligned}
$$

- Construct estimates of Z_θ and $Z_{\theta'}$ each iteration:
  - Take a sample set of configurations $\mathcal{X}$=(X_1,X_2,…,X_m)
  - Estimate Z_θ as

$$\tilde{Z}_\theta = \frac{1}{m} \sum_{x \in \mathcal{X}} exp(H_\theta(\mathbf{x}))$$

  - (and similarly for $Z_{\theta'}$)

Solution 1

- Nice result from theory of Approx. Bayesian Computation:
  - An MCMC process constructed in this way will have the correct stationary distribution ***provided we re-use our estimates***.

- Less nice properties:
  - You are getting an approximation.
  - This estimate of the normalizing constant is very noisy (most configurations are quite unlikely given θ (θ')

What do physicists do?

Statistical Science
1998, Vol. 13, No. 2, 163–185

Simulating Normalizing Constants: From Importance Sampling to Bridge Sampling to Path Sampling

Andrew Gelman and Xiao-Li Meng

Abstract. Computing (ratios of) normalizing constants of probability models is a fundamental computational problem for many statistical and scientific studies. Monte Carlo simulation is an effective technique, especially with complex and high-dimensional models. This paper aims to bring to the attention of general statistical audiences of some effective methods originating from theoretical physics and at the same time to explore these methods from a more statistical perspective, through establishing theoretical connections and illustrating their uses with statistical problems. We show that the *acceptance ratio method* and *thermodynamic*

What do physicists do?

- First insight: use the same set of sample configurations to estimate both Z_θ and $Z_{\theta'}$.
- But where should the samples come from?
  - In an MLE framework, sample configurations from $f(X \mid \hat{\theta})$
  - In Bayesian context we should presumably (if we could) be sampling θs from the posterior f(θ | X) and then sampling Xs from f(X | θ)….
  - But both are unknown

What do physicists do?

- Instead, when proposing a move from θ to θ' sample from f(X | θ) [or f(X | θ')?…]

- **Bridge sampling:** Use a "bridge" value of θ_B "halfway" between θ and θ'. Generate samples using θ_B (an MCMC process in itself!) and use those to estimate the ratio $Z_{\theta'} / Z_{\theta}$

  - May not work very well if θ and θ' are very different.

What do physicists do?

- **Path sampling:**
  - When proposing a move from θ to θ':
  - Build a "path" from θ to θ': $\theta^s \theta'^{(1-s)}$ for $0 \leq s \leq 1$
    - Generate samples using a range of s values spanning the path and construct your estimate of $Z_{\theta'} / Z_{\theta}$ from those. (Many MCMC processes!)

What do mathematicians do?

Biometrika (2006), **93,** 2, *pp.* 451–458

Printed in Great Britain

Miscellanea

An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants

BY J. MØLLER

Department of Mathematical Sciences, Aalborg University, Fredrik Bajers Vej 7G, 9220 Aalborg E, Denmark

jm@math.aau.dk

A. N. PETTITT, R. REEVES

School of Mathematical Sciences, Queensland University of Technology, GPO Box 2434, Brisbane, Queensland 4001, Australia

a.pettitt@qut.edu.au r.reeves@qut.edu.au

AND K. K. BERTHELSEN

Department of Mathematical Sciences, Aalborg University, Fredrik Bajers Vej 7G, 9220 Aalborg E, Denmark

kkb@math.aau.dk

What do mathematicians do?

- Path and bridge sampling provide approximate solutions.
- Møller et al. provide an exact solution.
- Introduce an auxiliary variable Y defined on the same state-space as X (in our context, Y is another (imaginary!) set of methylation configurations or proportions).
- Now mix over the space {θ, Y} as follows. At each MCMC iteration:

1. Propose a θ' according to some rule q(θ -> θ')
2. Propose a Y' according to f(X|θ).
3. Hastings ratio becomes:

$$
\begin{aligned}
h &= \min\left\{1, \frac{q(\theta' \to \theta)\ P(Y \mid \theta)\ P(\theta' \mid X)}{q(\theta \to \theta')\ P(Y' \mid \theta')\ P(\theta \mid X)}\right\} \\
&= \min\left\{1, \frac{q(\theta' \to \theta)\ \pi(\theta')\ [exp(H_\theta(Y))/\cancel{Z_\theta}]\ [exp(H_{\theta'}(X))/\cancel{Z_{\theta'}}]}{q(\theta \to \theta')\ \pi(\theta)\ [exp(H_{\theta'}(Y'))/\cancel{Z_{\theta'}}]\ [exp(H_\theta(X))/\cancel{Z_\theta}]}\right\}
\end{aligned}
$$

What do mathematicians do?

- There is no free lunch.
- You get an exact solution, but now you have to mix over Y as well. Møller et al. do this efficiently using *Perfect Sampling*.
- They show Ising model examples (on a 200x200 lattice) in which their methods works much better than existing software/methods in the context of exponential random graphs…..
- Perfect sampling is very tough to do, but if they can do it for Ising models on a 200x200 grid, then we ought to be able to get it done for 1 x #CpGsites grid for reasonably long regions. If not, resort to ordinary MCMC.
- This is going to be harder in a hierarchical model context (although the Perfect sampling piece would be parallelizable).

MCMC in R - “metrop” package

- MCMC1.Rmd on Github (in the MCMCpackage repo).
- Uses R’s built-in “metrop” package
- The example conducts a Bayesian analysis of the data ‘logit’ from R’s “mcmc” package.
- I have put the mcmc package doc file (“mcmc.pdf”) there as well

Summary of R MCMC library

- ‘metrop’ is a convenient way to implement simple MH-MCMC in R.
- Control the proposal step size (via scale), aiming to obtain an acceptance rate of around 20%
- You can thin the process out, using ‘nbatch' and ‘nspace’ to remove autocorrelation. (See Chapter 1 of “The Handbook of MCMC”, by Gelman et al., for a long discussion of whether you should sub-sample your data like this, or worry about ‘burn-in’.)

Exercises - non-examinable

1. Adapt the example from the MCMC1.Rmd document to run several MCMC chains for this problem and check convergence using the ‘coda’ package.

2. Use the metrop function to conduct an analysis of the dataset (QTLdata.txt) in that Repo
   - Genotype data for a set of 100 Single nucleotide polymorphisms, for 50 (inbred) individuals.
   - 1 row per individual; quantitative phenotype in last column.
   - The quantitative trait that depends upon one of those SNPs.
   - Find that SNP and estimate its ‘effect size’ (how much it changes the mean phenotype value).

3. **Use the metrop() function to implement the multivariate normal example from earlier today.**

END