Computer Science

CSC696H: Advanced Topics in Probabilistic Graphical Models

Implicit Model Inference

Prof. Jason Pacheco

Administrative Items

- I will be on travel the rest of this week
- No office hours this week
- There **is** class on Wednesday
- Caleb Dahlke will be helping out with discussion Wednesday
- Moyeen presenting ABC paper (Sunnaker et al. 2013)

Motivation for Monte Carlo Methods

- Now consider computing the expectation of a function $f(z)$ over $p(z)$.
- Recall that this looks like $E_{p(z)}[f] = \int_z f(z)p(z)dz$
- How can we approximate or estimate E[f]?

A bad plan…

Discretize the space where z lives into L blocks

Then compute $E_{p(z)}[f] \cong \frac{1}{L}\sum_{l=1}^{L} p(z)f(z)$

Scales poorly with dimension of Z

A better plan…

Given independant samples $z^{(l)}$ from $p(z)$

Estimate $E_{p(z)}[f] \cong \frac{1}{L}\sum_{l=1}^{L} f(z)$

Motivation for Monte Carlo Methods

- Real problems are typically complex and high dimensional.
- Suppose that we *could* generate samples from a distribution that is proportional to one we are interested in.
- Typically we want posterior samples,

$$p(z \mid \mathcal{D}) = \frac{p(z)p(\mathcal{D} \mid z)}{p(\mathcal{D})} \propto \widetilde{p}(z)$$

Unnormalized posterior

Don’t know marginal likelihood / normalizer

- Typically, $\widetilde{p}(z)$ is easier to evaluate (though not always)

Recall: Rejection Sampling

Assume

- Access to easy-to-sample distribution $q(z)$ ←
- Constant *k* such that $\widetilde{p}(z) \leq k \cdot q(z)$

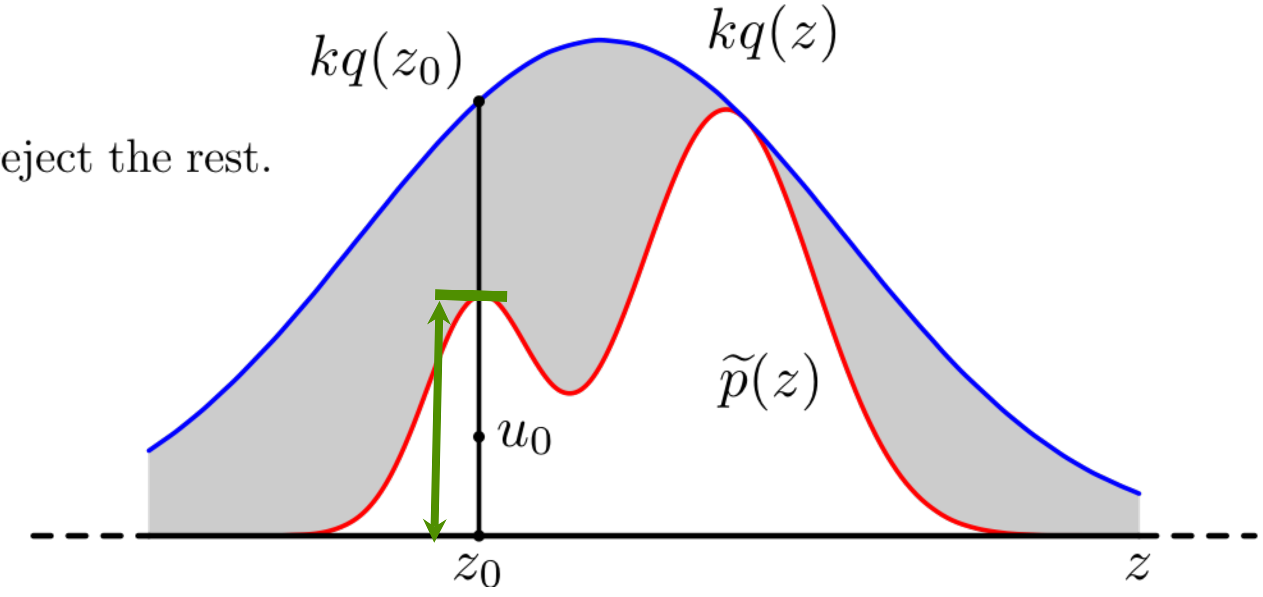
Proposal Distribution
Where we can use one of methods on previous slides to sample efficiently

Algorithm

1) Sample $q(z)$

2) Keep samples in proportion to $\dfrac{\tilde{p}(z)}{k \bullet q(z)}$ and reject the rest.

Example Use Gaussian proposal q to draw samples from multimodal distribution p



A Basic Monte Carlo Rejection Sampler

Goal: Given data $\mathcal{D}$ sample latent θ from posterior,

$$\theta \sim p(\theta \mid \mathcal{D})$$

Recall, **Bayes' Rule**:

$$p(\theta \mid \mathcal{D}) = \frac{\overset{\text{Prior}}{p(\theta)}\overset{\text{Likelihood}}{p(\mathcal{D}|\theta)}}{p(\mathcal{D}) \longleftarrow \text{Marginal Likelihood}}$$

A trivial Monte Carlo rejection sampler:

A1: Generate $\theta \sim p(\theta)$ from prior

A2: Accept θ with probability $h = p(\mathcal{D} \mid \theta)$

A3: Return to A1

A Basic Monte Carlo Rejection Sampler

```
A1: Generate $\theta \sim p(\theta)$ from prior
A2: Accept $\theta$ with probability $h = p(\mathcal{D} \mid \theta)$
A3: Return to A1
```

- It's trivial to show that this has the correct *target distribution,*

$$\theta \sim p(\theta \mid \mathcal{D})$$

- Special case of a Rejection Sampler with proposal $\theta \sim p(\theta)$

- In general, find an upper bound $c \geq p(\mathcal{D} \mid \theta)$ and accept with prob. h/c

What are some issues with this sampler?

Problems with the Basic Rejection Sampler

Problem 1: The prior is not a good proposal in general, since it is often very different from the posterior:

$$p(\theta) \neq p(\theta \mid \mathcal{D})$$

Problem 2: To compute the acceptance we need to be able to *evaluate the likelihood*:

$$h = p(\mathcal{D} \mid \theta)$$

Main Point: Many likelihood models are easily defined via *simulation* but cannot be explicitly evaluated.

- Easy to simulate new data: $\mathcal{D}' \sim p(\cdot \mid \theta)$
- Can't evaluate likelihood at specific data / parameter: $\cancel{p(\mathcal{D} \mid \theta)}$

Implicit vs. Explicit Models

Typically we know, both, the **prior** and **likelihood** of the joint,

$$p(\theta, \mathcal{D}) = p(\theta)p(\mathcal{D} \mid \theta)$$

- We call this an **explicit model**
- An **implicit model** lacks a closed-form joint
- Models are usually implicit because we don't know the likelihood

Two common reasons for implicit likelihood:

1) Need to integrate nuisance variables,

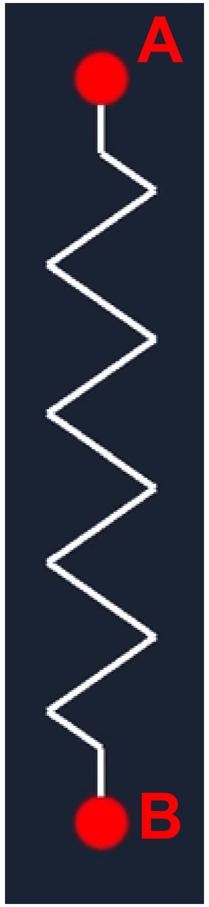
Can address this with standard inference

$$p(\mathcal{D} \mid \theta) = \int p(\theta, \eta)p(\mathcal{D} \mid \eta, \theta)\, d\eta$$

2) Likelihood is based on simulation

Topic of this paper

Example: Mass-Spring Simulation



Represents *mass* and *elasticity* of a soft body using:

A, B : Two mass points

κ_s : Spring stiffness

L_0: Rest length

κ_d: Damping factor

Subset of these represent parameters θ

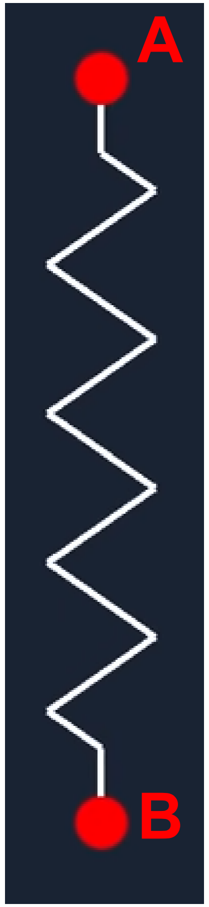
Simulate by using Hooke's Law:

$$F_s = \kappa_s \underbrace{\left(|B - A| - L_0\right)}_{\text{Deviation from rest length}}$$

Force on Spring

YT: Gonkee: https://youtu.be/kyQP4t_wOGI

Example: Mass-Spring Simulation



Need to add *damping force* to avoid never-ending simulation,

$$F_d = \left(\frac{B-A}{|B-A|}\right) \cdot (v_B - v_A)\kappa_d$$

Unit vector of Position A-to-B

Motion Vectors

Damping Factor

Total force is sum of spring and damping forces,

$$F_t = F_s + F_d$$

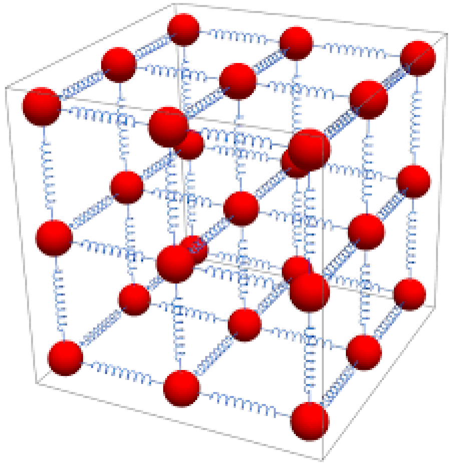
Can easily simulate this in CPU using numerical integration, e.g. Euler's method,

$$v(t) = v(t-1) + \frac{F_t \Delta t}{m} \qquad x(t) = x(t-1) + v(t)\Delta t$$

YT: Gonkee: https://youtu.be/kyQP4t_wOGI

Example: Lattice-Spring Model

Extend mass-spring to multiple masses / springs



- Simulating data $\mathcal{D}$ from parameters θ is easy
- Can simulate complicated physics like

  Soft-body Tetris
- Simple setting is deterministic
- Simulation is **much easier** than writing down a function tying inputs to outputs,

$$\mathcal{D} = f(\theta)$$

- Can easily add noise to make random, but can’t write down likelihood,

$$p(\mathcal{D} \mid \theta)$$

Likelihood-Free Monte Carlo

```
B1: Generate $\theta \sim p(\theta)$ from prior
B2: Simulate $\mathcal{D}'$ from model with input $\theta$
B3: Accept $\theta$ if $\mathcal{D}' = \mathcal{D}$ ; Return to B1
```

- Unlike rejection sampler, never need to evaluate likelihood
- Probability of acceptance is proportional to $p(\mathcal{D})$
- Prohibitively low acceptance for high-dimensional data
- **Idea** Make acceptance criteria weaker… accept within some distance:

$$\rho(\mathcal{D}, \mathcal{D}') \leq \epsilon$$

Likelihood-Free Monte Carlo

C1: Generate $\theta \sim p(\theta)$ from prior

C2: Simulate $\mathcal{D}'$ from model with input θ

C3: Calculate distance $\rho(\mathcal{D}', \mathcal{D})$

C4: Accept θ if $\rho(\mathcal{D}', \mathcal{D}) \leq \epsilon$; Return to C1

- Will have higher acceptance than Algorithm B
- Target distribution is approximation of true posterior,

$$p(\theta \mid \rho(\mathcal{D}, \mathcal{D}') \leq \epsilon) \approx p(\theta \mid \mathcal{D})$$

- This still won’t work in high-dimensional data…too many rejections
- **Idea** Test a *statistic S* instead…

Likelihood-Free Monte Carlo

```
D1: Generate $\theta \sim p(\theta)$ from prior
D2: Simulate $\mathcal{D}'$ from model with input $\theta$
D3: Compute statistic $S'$ of $\mathcal{D}'$
D4: Calculate distance $\rho(\mathcal{S}', \mathcal{S})$
D5: Accept $\theta$ if $\rho(\mathcal{S}', \mathcal{S}) \leq \epsilon$ ; Return to D1
```

- Typically higher acceptance rate than Algorithm C
- Target distribution is an even rougher approximation of true posterior,

$$p(\theta \mid \rho(S, S') \leq \epsilon) \approx p(\theta \mid \mathcal{D})$$

- Finding statistics that make this a good approximation is hard
- Standard statistics: mean, median, min, max, etc.

Likelihood-Free Inference So Far

Draw sample from prior $\theta \sim p(\theta)$:

- Basic rejection sampling, requires likelihood (Alg. A)
- Accept sample only if simulated data matches real (Alg. B)
- Accept sample if data are *close enough* (Alg. C)
- Accept sample if *statistics* are close enough (Alg. D)

➢ Prior distribution is bad proposal in general

➢ Posterior is typically very different from prior

➢ Need a better proposal…

Metropolis-Hastings MCMC

E1: Propose move $\theta' \sim q(\theta' \mid \theta)$

E2: Calculate,

$$h = \min\left(1, \frac{p(\mathcal{D}|\theta')p(\theta')q(\theta|\theta')}{p(\mathcal{D}|\theta)p(\theta)q(\theta'|\theta)}\right)$$

E3: Move to θ' with probability h, else stay at θ; Return to E1

- MCMC gradually adjusts proposal towards posterior
- Stationary distribution of Markov chain is the true posterior
- But, M-H acceptance ratio requires evaluation of likelihood ratio

Approximating the Likelihood Ratio

M-H acceptance requires computing the likelihood ratio:

$$\frac{p(\mathcal{D} \mid \theta')}{p(\mathcal{D} \mid \theta)}$$

- Approximate each term by simulating B datasets, $\mathcal{D}_1, \ldots, \mathcal{D}_B$
- Then compute the empirical mean:

$$\hat{p}(\mathcal{D} \mid \theta) = \frac{1}{B} \sum_{j=1}^{B} I(\mathcal{D}_j = \mathcal{D})$$

- Where I(.) is the Kroenecker delta
- A trivial case is when B=1

MCMC Without Likelihoods

```
F1: Propose move $\theta' \sim q(\theta' \mid \theta)$
F2: Generate $\mathcal{D}'$ using inputs $\theta'$
F3: If $\mathcal{D}' = \mathcal{D}$ goto F4 otherwise stay at $\theta$
F4: Calculate,
        $h = \min\left(1, \frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)}\right)$
F5: Move to $\theta'$ with probability h, else stay
    at $\theta$; Return to F1
```

Theorem in paper proves stationary distribution is still true posterior

MCMC Without Likelihoods

- Just as in Algorithm B almost all samples will be rejected
- Especially if data are high-dimensional…

To improve acceptance rate, continue if data is *close enough:*

F3’: If $\rho(\mathcal{D}, \mathcal{D}') \leq \epsilon$ goto F4 otherwise stay at θ

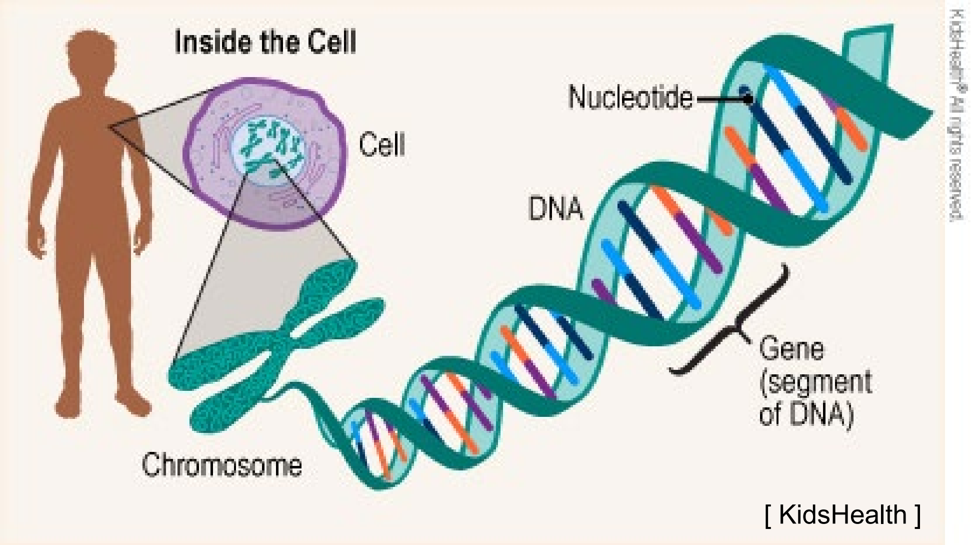
Or close enough with respect to a *statistic*:

F3’’: If $\rho(S, S') \leq \epsilon$ goto F4 otherwise stay at θ

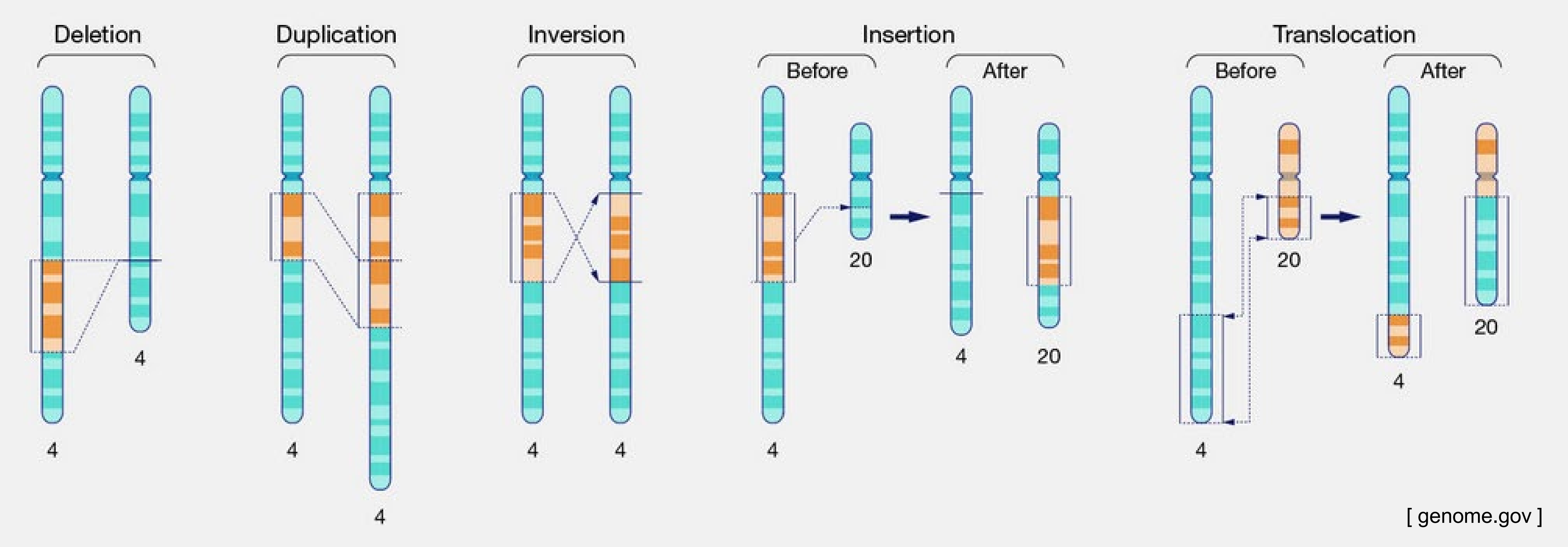
These are same changes made to rejection sampling, but for Metropolis-Hastings

Basics of DNA and Mutations

- Double-helix of nucleotide strands
- 4 nucleotides (A, C, G, T)
- Pairings A-T, G-C form double helix
- Replication of DNA can cause mutations
- Usually, mutations caught and discarded

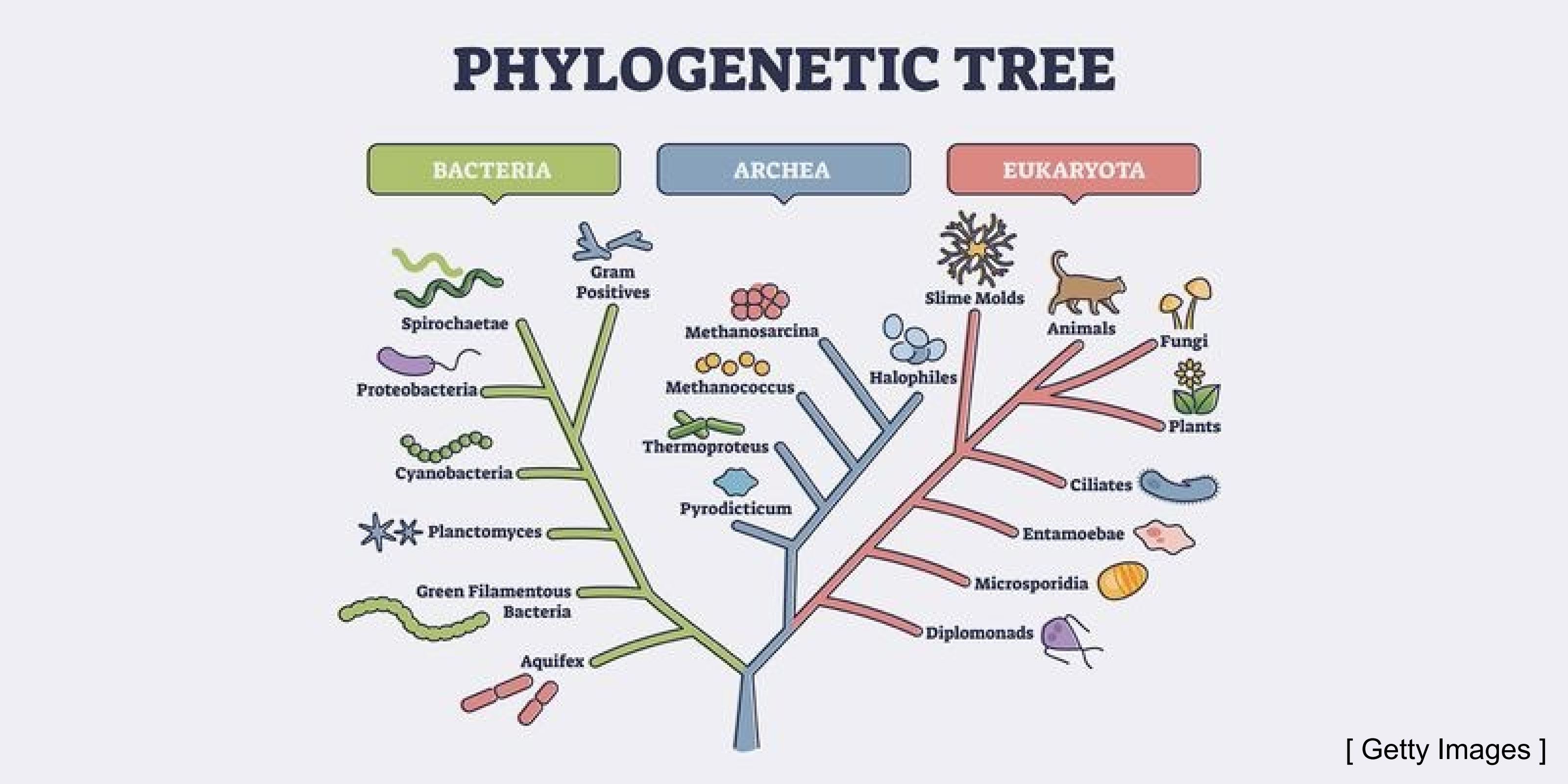


[KidsHealth]



[genome.gov]

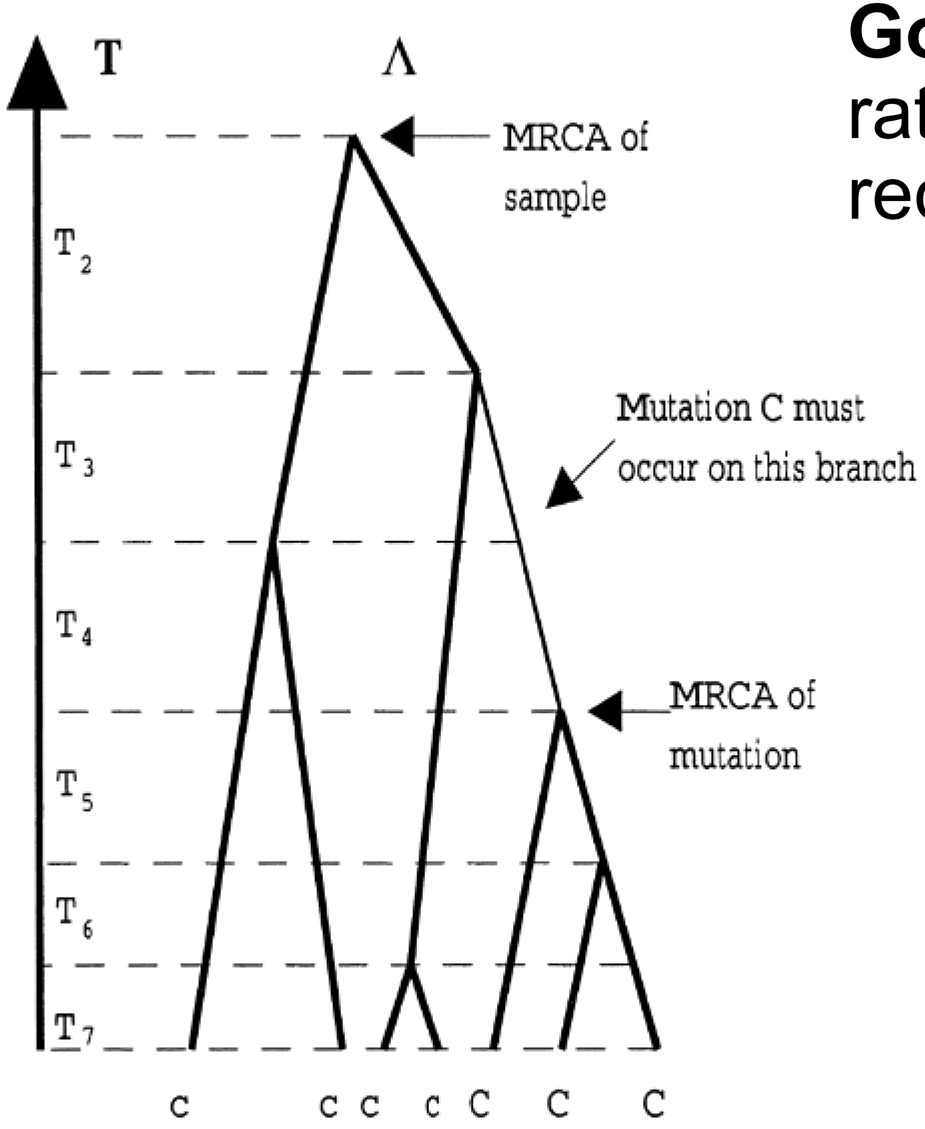
Population Genetics



[Getty Images]

- Sometimes mutations persist and are inherited by later generations
- This leads to divergence of populations
- Then to different species, etc.
- **Question** Given some DNA samples, what is the most recent ancestor?

Example: Population Genetics



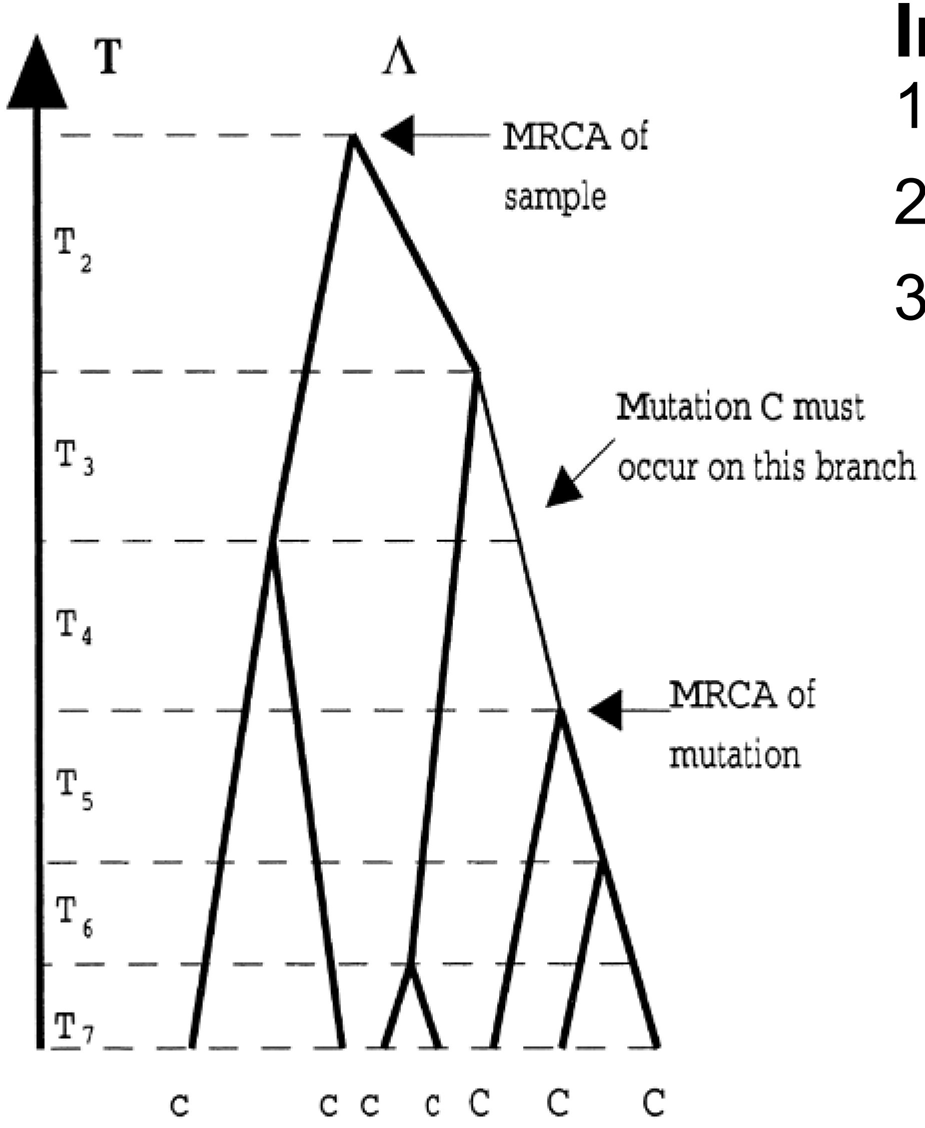
[Markovtsova et al. (2000)]

Goal Given DNA samples determine mutation rates, times of mutation events, and most recent common ancestor (MRCA)

Coalescent Model (Simple Version)

- Assumes random mating of population size N
- Sample n < N sequences at present day
- Run time backward in units of N/σ^2 generations where σ^2 is variance of num. of offspring in 1 generation
- At time T_j sample has j ancestors
- $T_j \sim \text{Exponential}(j(j-1)/2)$
- Stop when a single line of ancestry remains

Example: Population Genetics



[Markovtsova et al. (2000)]

Implementing Algorithm F

1) Propose mutation rate

2) Generate new tree topology Λ and set of mutations

3) Compare to samples do M-H acceptance

Naïve implementation leads to low acceptance…

- Augment state-space with tree topology Λ and times of coalescence on the topology
- **Intuition** Including more information in state space allows more local moves in that space and improves acceptance rate
- I.e. when we find a good state we make small changes that are even better
- **Tradeoff** Larger state space, smaller moves

Example: Population Genetics

Additional algorithm details…

- Characterize mutations by:
  - Time they occur (i.e. branch they happen on)
  - Their location on the genome
- Include number of mutations between two coalescent events
- Location of mutations chosen uniformly among tree branches during simulation

Marjoram et al. (2003) claim this is the least information needed to see reasonable acceptance

Example: Population Genetics

Update Process (proposal step in M-H)

- Update topology of tree (details in Markovstova et al. [2000])
- Update times between coalescent events by adding Gaussian noise
- Update mutation rate by adding uniform random noise
- New mutation rate and times define Poisson RV of number of mutations between pairs of coalescence events
- For new mutation choose location in genome and tree uniformly
- If number of mutations decreases randomly select some mutations and erase them

Results: Marjoram et al. 2003

Dataset / Methodology

- Sample n=63 sequences
- From Nuu-chah-nulth (Nootka) indigenous people of Pacific NW
- Sequences are 360 base pairs (bp) long
- Observed base frequencies $(\pi_A, \pi_G, \pi_C, \pi_T) = (0.330, 0.112, 0.337, 0.221)$
- H=28 distinct sequences (haplotypes)
- V=26 base positions showing variation
- Inference on (rescaled) mutation param θ and height of tree T
- Using Algorithm F, with previously discussed modifications

Results: Marjoram et al. 2003

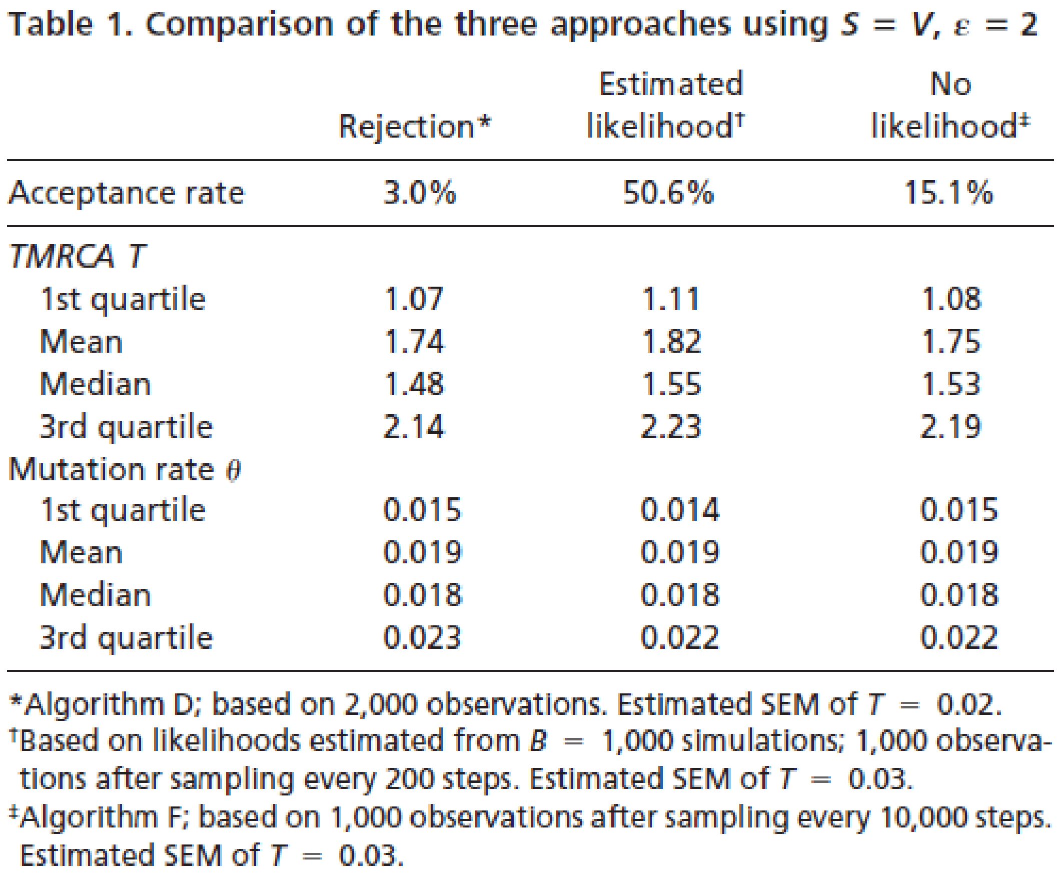


Table 1. Comparison of the three approaches using $S = V$, $\varepsilon = 2$

	Rejection*	Estimated likelihood†	No likelihood‡
Acceptance rate	3.0%	50.6%	15.1%
TMRCA T			
1st quartile	1.07	1.11	1.08
Mean	1.74	1.82	1.75
Median	1.48	1.55	1.53
3rd quartile	2.14	2.23	2.19
Mutation rate θ			
1st quartile	0.015	0.014	0.015
Mean	0.019	0.019	0.019
Median	0.018	0.018	0.018
3rd quartile	0.023	0.022	0.022

*Algorithm D; based on 2,000 observations. Estimated SEM of T = 0.02.

†Based on likelihoods estimated from B = 1,000 simulations; 1,000 observations after sampling every 200 steps. Estimated SEM of T = 0.03.

‡Algorithm F; based on 1,000 observations after sampling every 10,000 steps. Estimated SEM of T = 0.03.

- Compare rejection, estimated likelihood, and likelihood-free MCMC
- Use summary stats S=V
- Data accepted if $|S - V| \leq \epsilon$
- First compare with $\epsilon = 2$

Observations

- Methods produce comparable T
- Comparable mutation rate
- Very different acceptance rates

Results: Marjoram et al. 2003

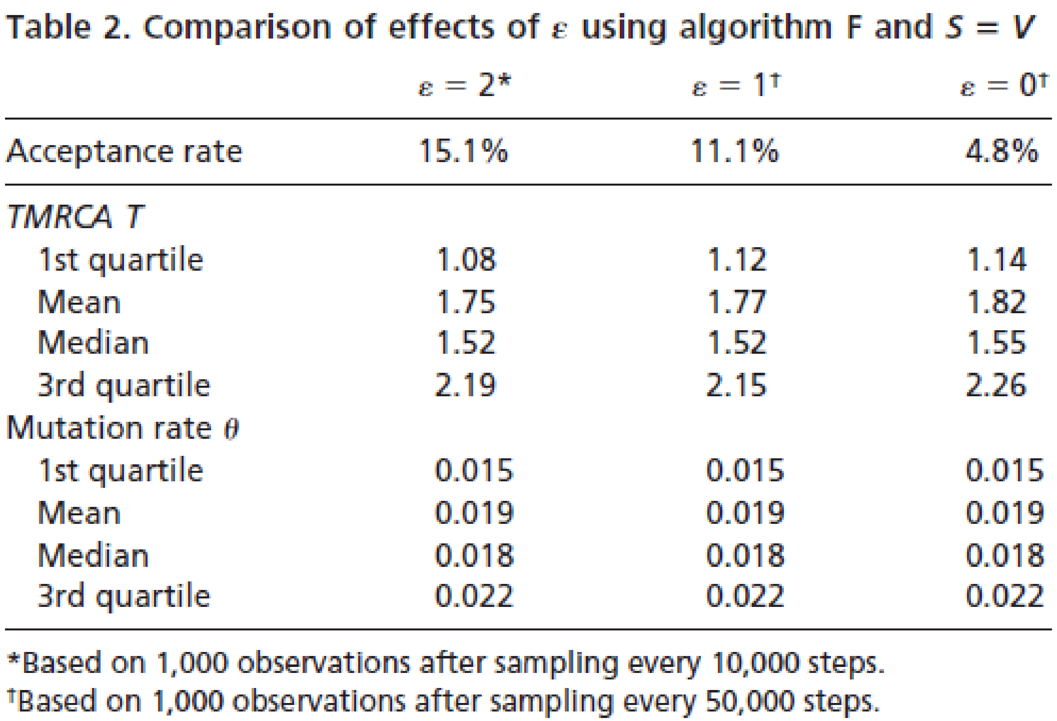


Table 2. Comparison of effects of ε using algorithm F and $S = V$

	$\varepsilon = 2$*	$\varepsilon = 1$†	$\varepsilon = 0$†
Acceptance rate	15.1%	11.1%	4.8%
TMRCA T			
1st quartile	1.08	1.12	1.14
Mean	1.75	1.77	1.82
Median	1.52	1.52	1.55
3rd quartile	2.19	2.15	2.26
Mutation rate θ			
1st quartile	0.015	0.015	0.015
Mean	0.019	0.019	0.019
Median	0.018	0.018	0.018
3rd quartile	0.022	0.022	0.022

*Based on 1,000 observations after sampling every 10,000 steps.

†Based on 1,000 observations after sampling every 50,000 steps.

Look at varying ϵ for MCMC

- “Under coalescent prior, mean heigh of tree is 1.97; posterior means do not differ from this”
- Surprisingly, $\epsilon = 0$ still has non-negligible acceptance
- Acceptance rate pretty low overall

Results: Marjoram et al. 2003

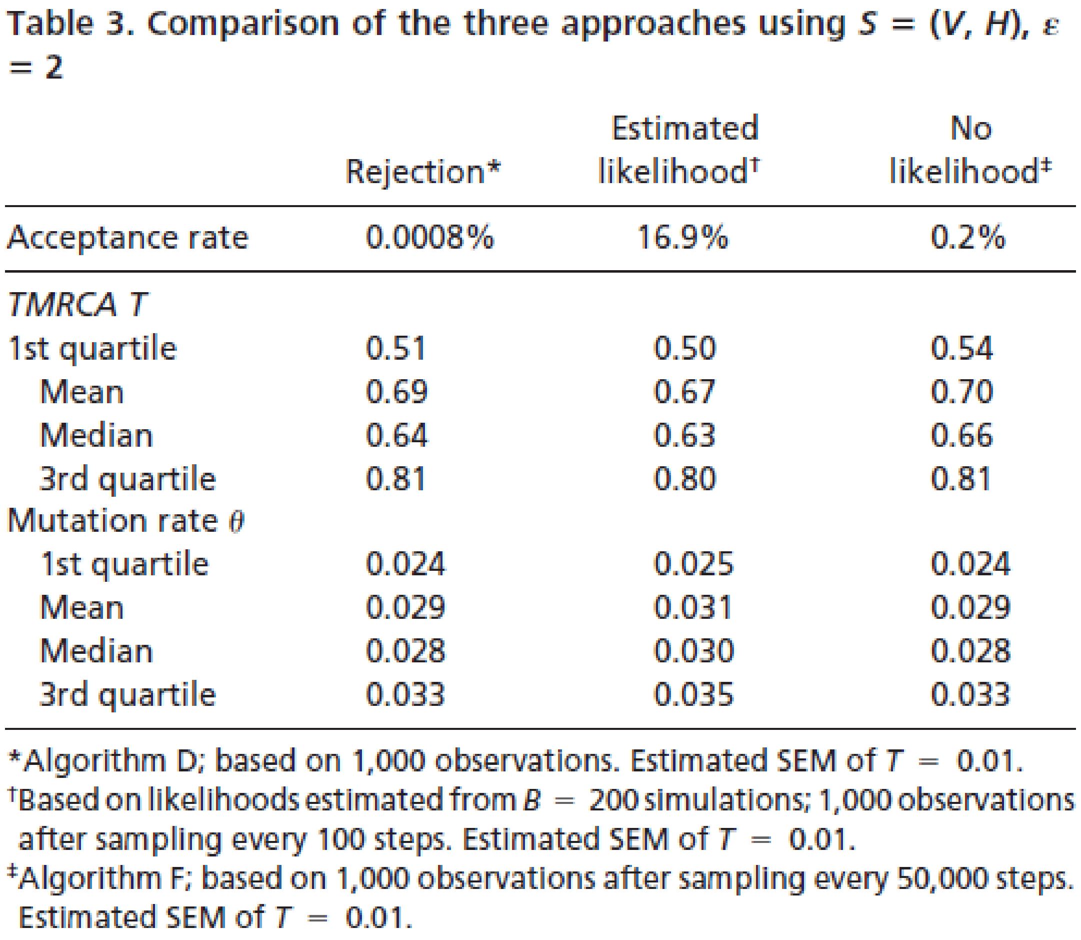


Table 3. Comparison of the three approaches using S = (V, H), ε = 2

	Rejection*	Estimated likelihood†	No likelihood‡
Acceptance rate	0.0008%	16.9%	0.2%
TMRCA T			
1st quartile	0.51	0.50	0.54
Mean	0.69	0.67	0.70
Median	0.64	0.63	0.66
3rd quartile	0.81	0.80	0.81
Mutation rate θ			
1st quartile	0.024	0.025	0.024
Mean	0.029	0.031	0.029
Median	0.028	0.030	0.028
3rd quartile	0.033	0.035	0.033

*Algorithm D; based on 1,000 observations. Estimated SEM of T = 0.01.

†Based on likelihoods estimated from B = 200 simulations; 1,000 observations after sampling every 100 steps. Estimated SEM of T = 0.01.

‡Algorithm F; based on 1,000 observations after sampling every 50,000 steps. Estimated SEM of T = 0.01.

Authors state “Estimated likelihood method is at the edge of feasibility…”

Use stats S=(V,H) and $\epsilon = 2$ *accept if:*

$$|H - 28| \leq \epsilon \qquad |V - 26| \leq \epsilon$$

- Using more complicated MCMC of Markovstova et al. (2000) mean height estimated at 0.68
- Using S=(V,H) yields results much closer to this estimate
- Rejection sampler essentially useless
- Likelihood estimation still higher acceptance, and closer estimate to “true” result

Results: Marjoram et al. 2003

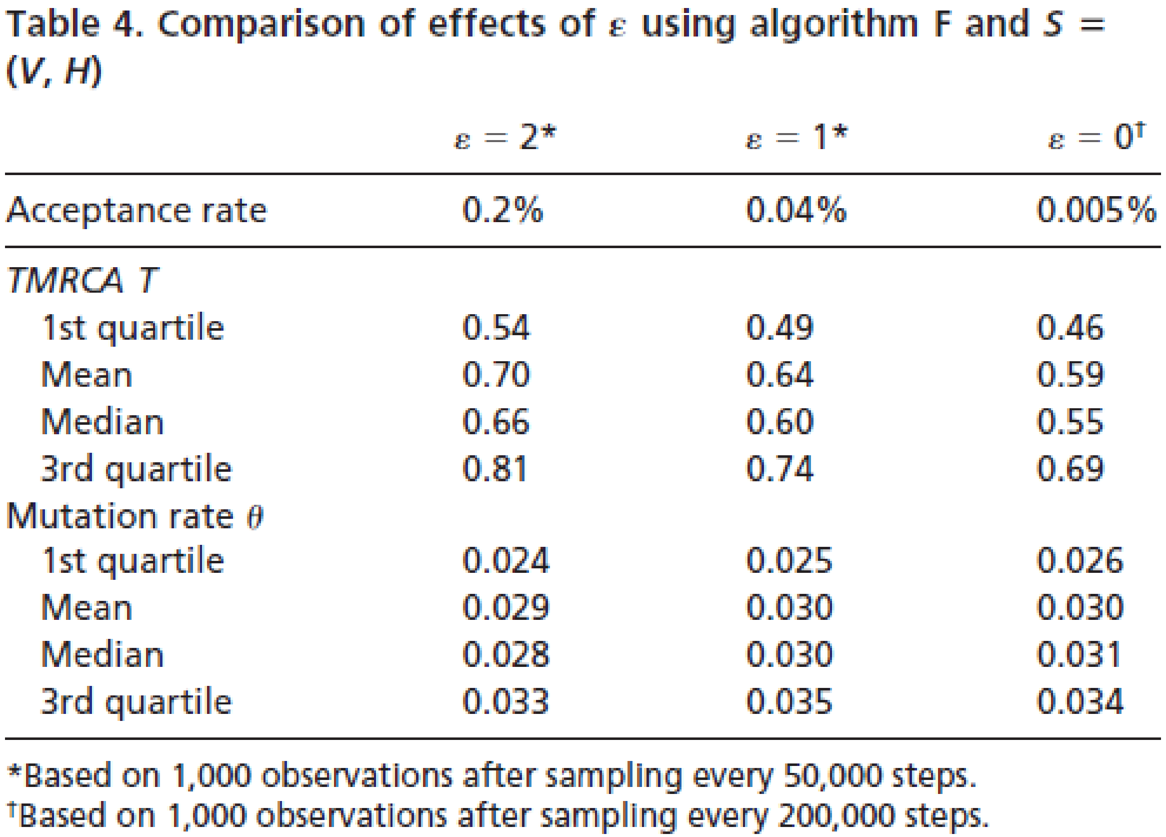


Table 4. Comparison of effects of ε using algorithm F and $S = (V, H)$

	$\varepsilon = 2$*	$\varepsilon = 1$*	$\varepsilon = 0$†
Acceptance rate	0.2%	0.04%	0.005%
TMRCA T			
1st quartile	0.54	0.49	0.46
Mean	0.70	0.64	0.59
Median	0.66	0.60	0.55
3rd quartile	0.81	0.74	0.69
Mutation rate θ			
1st quartile	0.024	0.025	0.026
Mean	0.029	0.030	0.030
Median	0.028	0.030	0.031
3rd quartile	0.033	0.035	0.034

*Based on 1,000 observations after sampling every 50,000 steps.

†Based on 1,000 observations after sampling every 200,000 steps.

Varying MCMC threshold…

- Overall low acceptance
- Higher threshold yields more accurate estimates (compared to "truth")
- Not feasible below 2.0
- So, it works… with some caveats… and tuning… definitely not an out-of-the-box solution