Stochastic models for simulation and inference

Molecular Epidemiology of Infectious Diseases

Lecture 11

March 30th, 2026

The road ahead

Last week: Modeling epidemic dynamics with SIR models

This week: Stochastic models for simulation and inference

Next week: putting everything together with phylodynamic modeling

Why include stochasticity in our models?

Why include stochasticity?

- To account for true randomness in a process (e.g. rolling dice)
- To account for uncertainty/heterogeneity in a process
- To capture realistic amounts of variability in observed data

Accounting for true randomness

Some processes might be truly random or stochastic (e.g. electrons in quantum theory), but this is scientifically and philosophically debatable.

Rather, it is often convenient to model a physical process as random.

Example: we model the outcome of rolling dice as a random variable because the physics required to model this deterministically is very complicated and would require us to know a lot of information.

"I, in any case, am convinced He [God] **does not play dice** with the **universe**."

Einstein (1926)

Accounting for uncertainty

We often model processes at much smaller and larger scales than we are interested in using ***phenomenological models***.

Example: The transmission rate of a foliar pathogen from one plant to another could be modeled mechanistically if we knew a lot about the number of spores on each leaf, spore viability, relative humidity, wind speeds, ect.

We generally don't have access to this detailed information so we take into account our uncertainty about individual outcomes using stochastic models.

Accounting for uncertainty

We often model processes at much smaller and larger scales than we are interested in using ***phenomenological models***.

Example: The transmission rate of a foliar pathogen from one plant to another could be modeled mechanistically if we knew a lot about the number of spores on each leaf, spore viability, relative humidity, wind speeds, ect.

We generally don't have access to this detailed information so we take into account our uncertainty about individual outcomes using stochastic models.

In this sense, stochasticity is a hedge against our own ignorance.

Capturing variability in data

Empirical data often include much more heterogeneity and variability than our simple deterministic models would suggest.

When performing statistical inference, including stochasticity in our models allows us to fit models that are flexible enough to account for this variability.

When making predictions/forecasting, simulating data often requires us to add randomness in order to generate an ensemble of different possible outcomes.

Why include stochasticity?

- To account for true randomness in a process (e.g. rolling dice)
- To account for uncertainty/heterogeneity in a process
- To capture realistic amounts of variability in observed data

How do we include stochasticity in our models?

Different types of stochasticity

Observational noise: error in our observations that don't actually affect the process under study (e.g. misreporting of infections).

Example: The true frequency of infected individuals might be ***p***, but we observe a noisy estimate of this frequency if we only sample ***n*** individuals..

The observed number of infections **k** would follow a binomial distribution:

$$\Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Different types of stochasticity

Observational noise: error in our observations that don't actually affect the process under study (e.g. misreporting of infections).

Process noise: randomness in a process that changes the dynamic behavior of the system under study.

Different types of stochasticity

Observational noise: error in our observations that don't actually affect the process under study (e.g. misreporting of infections).

Process noise: randomness in a process that changes the dynamic behavior of the system under study.

> *Environmental noise:* internal or external perturbations (e.g. climatic factors like relative humidity affecting transmission rates).
>
> *Demographic stochasticity:* randomness in the timing and outcome of individual events at the individual level (e.g. the time at which a given individual is born and dies).

Modeling environmental noise

Environmental stochasticity is normally modeled as random noise due to external factors entering our model:

For dynamical systems like SIR models, we can add noise to the rates of change using stochastic differential equations:

$$\frac{dI}{dt} = \beta SI - \gamma I + \xi\beta SI$$

ξ is a “noise” increment, generally a random Normal variate.

Modeling demographic stochasticity

Demographic stochasticity requires us to consider randomness in the timing and outcome of individual events.

Individuals are treated as discrete (whole) units, so that the number of individuals is always an integer.

We can think about individual events as resulting from reactions involving one or more individuals.

Modeling demographic stochasticity

For a stochastic SIR model, we have transmission events resulting from the interaction of susceptible and infected hosts:

$$S + I \xrightarrow[\beta]{} 2I$$

Recovery events convert infecteds into recovered hosts:

$$I \xrightarrow[\gamma]{} R$$

Event driven simulation

If we want to simulate with demographic stochasticity for a population with many individuals and competing events/reactions, we can use the following general approach:

1. Simulate or draw the time of the next event.
2. Choose type of event that occurs at that time
3. Update the time and the population states to reflect any changes in the number of individuals of each type.

Step 1: Finding the next event time

To draw the time of the next event we:

a) First compute the rate of each reaction or event type:

$$\lambda_T = \beta SI \qquad \lambda_R = \gamma I$$

b) Sum the individual rates to find the total rate:

$$\lambda_{Total} = \lambda_T + \lambda_R$$

c) Draw the time to the next event from an exponential distribution:

$$\text{Time to next event} = \Delta_t \sim \text{Exponential}(\lambda_{Total})$$

Step 2: Choose the event type

In order to choose what type of event occurs we:

a) Find the relative probability of each event type:

$$P(\text{Transmission}) = \frac{\lambda_T}{\lambda_T + \lambda_R} \qquad P(\text{Recovery}) = \frac{\lambda_R}{\lambda_T + \lambda_R}$$

b) Choose the event type according to the relative probability of each event (i.e. a single draw from a multinomial distribution).

Step 3: Update time and states

Update the system by moving ahead to the next event time and updating the appropriate population states.

For example, if the next event is a transmission event we set:

$$t \to t + \Delta t$$

$$S \to S - 1$$

$$I \to I + 1$$

Event driven simulation

If we want to simulate with demographic stochasticity for a population with many individuals and competing events/reactions, we can use the following general approach:

1. Simulate or draw the time of the next event.
2. Choose type of event that occurs at that time
3. Update the time and the population states to reflect any changes in the number of individuals of each type.

This process is repeatedly iterated until some *end condition* is met.

The general SSA

The general stochastic simulation algorithm (SSA) can be used to simulate any continuous time, discrete state Markov process. It is sometimes referred to as the Gillespie algorithm (Gillespie, 1977).

1.) Label all events E_1, E_2, ... E_N
2.) Compute the rate λ_i of each event type *i*
3.) Compute the total rate λ_{Total} by summing the individual rates.
4.) Draw the next event time from an exponential distribution with rate λ_{Total}
5.) Choose event type *E* according to the relative event probabilities
6.) Update time and states

The stochastic simulation algorithm can be used to simulate just about everything!

Applications of the SSA

We can apply the SSA to simulate:

- Epidemic dynamics with stochasticity
- Phylogenetic and transmission trees
- Molecular evolution for sequence data

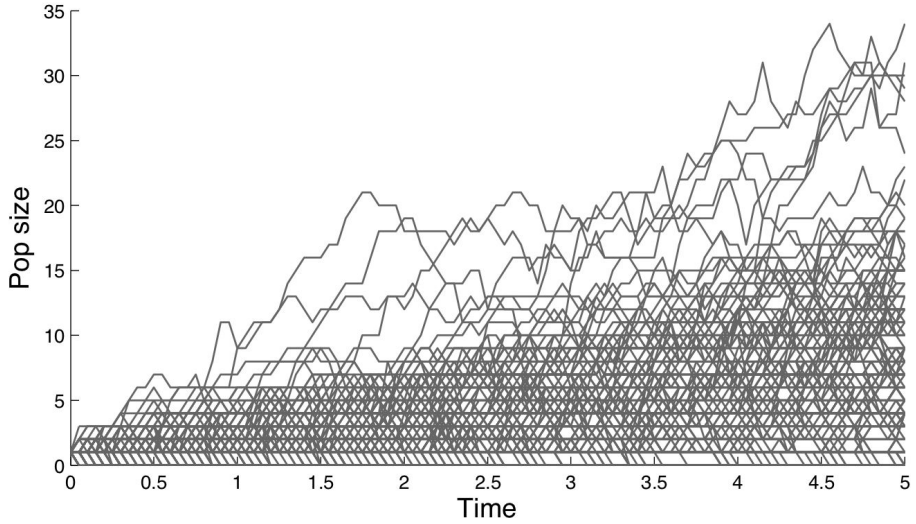
Applications of the SSA

We can apply the SSA to simulate:

- Epidemic dynamics with stochasticity
- Phylogenetic and transmission trees
- Molecular evolution for sequence data

Stochastic epidemic dynamics

Stochasticity is especially important when populations are small and individual events have a large effect on the overall dynamics.



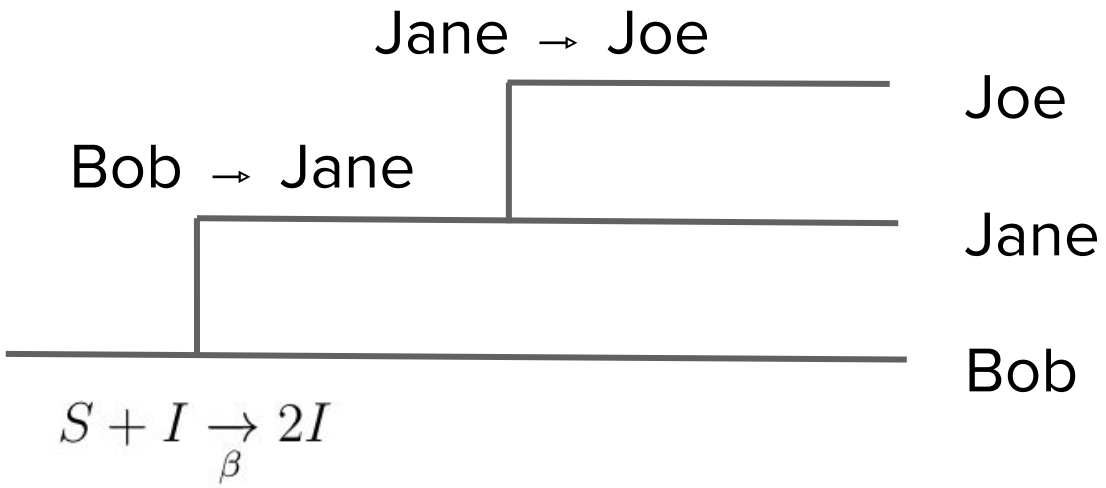
Applications of the SSA

We can apply the SSA to simulate:

- Epidemic dynamics with stochasticity
- Phylogenetic and transmission trees
- Molecular evolution for sequence data

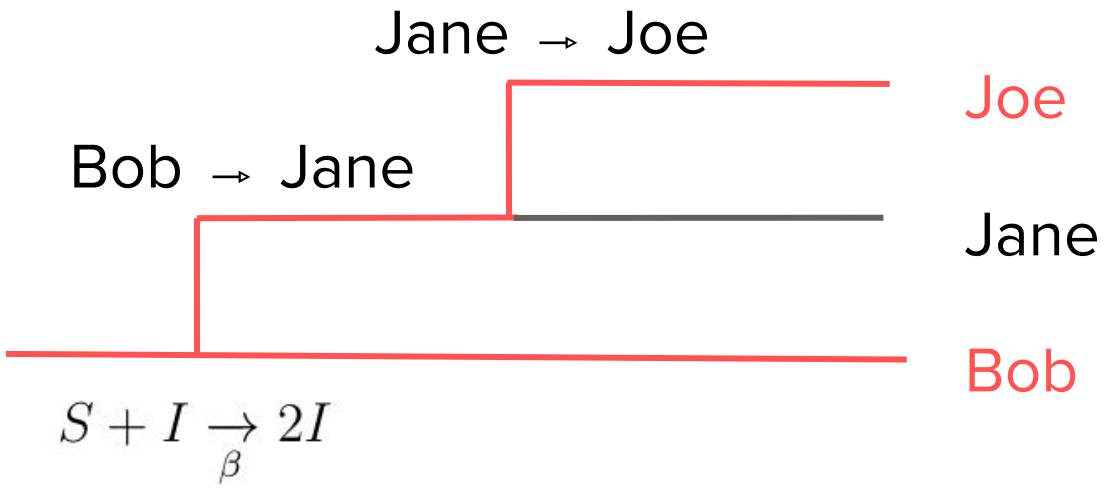
Simulating transmission trees

We can simulate the transmission tree while simulating epidemic dynamics under the SSA by recording the ancestry of the population in terms of parent child relationships:



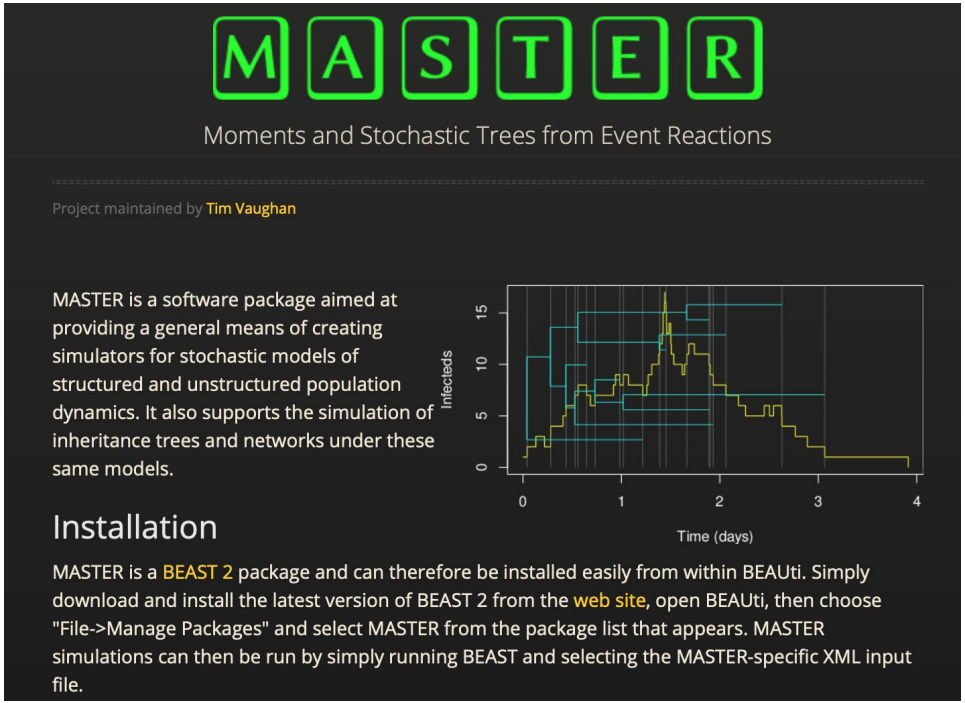
Simulating transmission trees

We can then trace the ancestry of sampled individuals backwards through time to obtain the transmission tree:



Simulating trees and epidemics

MASTER is a BEAST 2 package that allows for efficient stochastic simulation of population dynamics and trees.



Vaughan and Drummond (MBE, 2013)

Simulating trees and epidemics

MASTER is a BEAST 2 package that allows for efficient stochastic simulation of population dynamics and trees.

Events are specified as “reactions” in an input XML file.

```
<run spec='Trajectory'
     simulationTime='50'>

    <model spec='Model' id='model'>
          <population spec='Population' id='S' populationName='S'/>
          <population spec='Population' id='E' populationName='E'/>
          <population spec='Population' id='I' populationName='I'/>
          <population spec='Population' id='R' populationName='R'/>

          <reaction spec='Reaction' reactionName="Infection" rate="0.05">
            S + I -> I + E
          </reaction>
          <reaction spec='Reaction' reactionName="Incubation" rate="0.4">
            E -> I
          </reaction>
          <reaction spec='Reaction' reactionName="Recovery" rate="0.2">
            I -> R
          </reaction>
    </model>
```

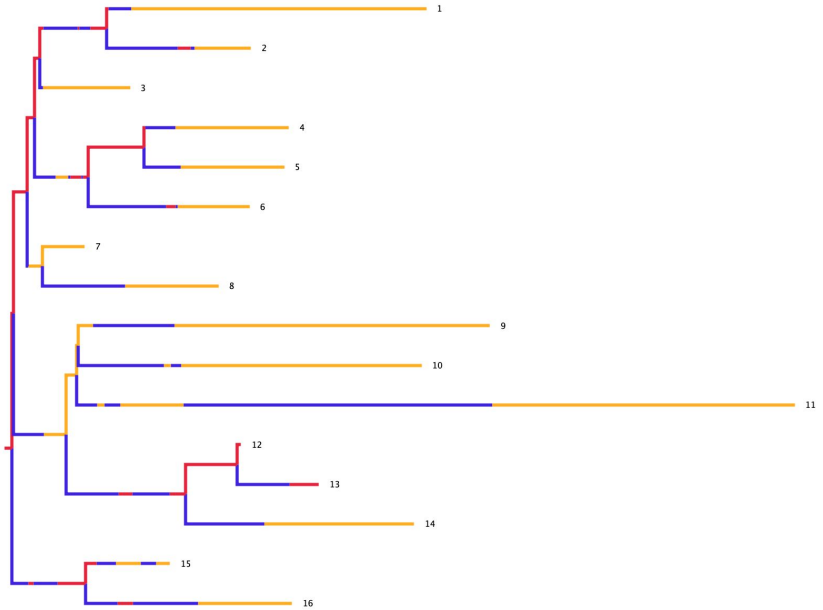
Vaughan and Drummond (MBE, 2013)

Simulating trees and epidemics

MASTER is a BEAST 2 package that allows for efficient stochastic simulation of population dynamics and trees.

Events are specified as “reactions” in an input XML file.

Epidemic dynamics and trees are saved as output.



Vaughan and Drummond (MBE, 2013)

Packages for simulating trees

Popular choices include:

- MASTER https://tgvaughan.github.io/MASTER/
  - Forward-time stochastic simulations of epidemics and trees in BEAST 2
  - Very flexible model specification (e.g. easy to add different types of hosts)
- msprime https://tskit.dev/msprime
  - Backwards-time coalescent simulator
  - Allows for different (deterministic) demographic histories
  - Great for simulating ARGs and sequences with recombination
- SLiM https://messerlab.org/slim/
  - Forward-time simulator of trees and sequence data
  - Designed for Wright-Fisher simulations but very extensible to more complex models.
  - Great for simulating non-neutral models with selection. Fun GUI!

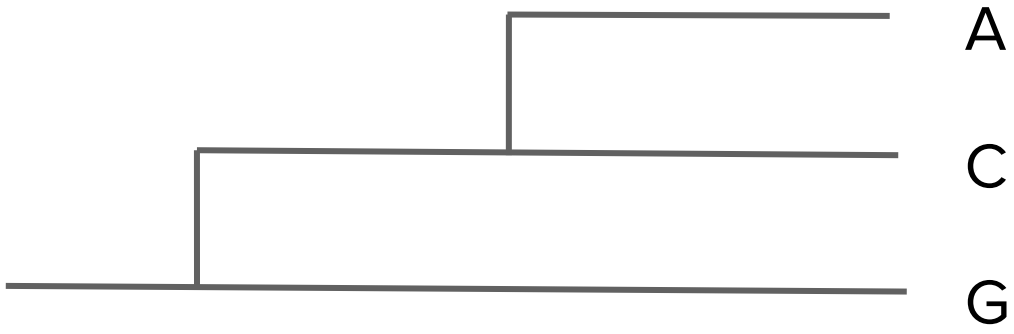
Applications of the SSA

We can apply the SSA to simulate:

- Epidemic dynamics with stochasticity
- Phylogenetic and transmission trees
- Molecular evolution for sequence data

Simulating sequences on trees

We can use the SSA to simulate molecular evolution along each lineage in a phylogeny to obtain simulated sequence data at the tips:

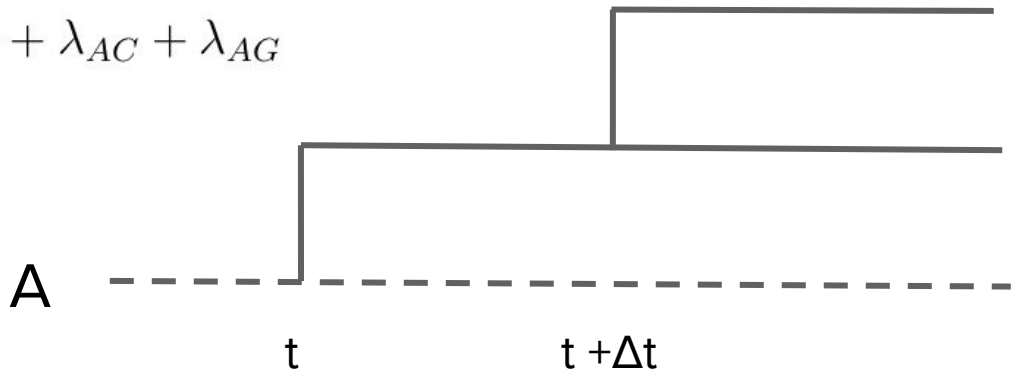


Simulating sequences on trees

Step 1: Compute the rate at which all substitution events occur to find the total mutation rate and then draw time to first mutation:

$$\text{Time to next event} = \Delta_t \sim \text{Exponential}(\lambda_{Total})$$

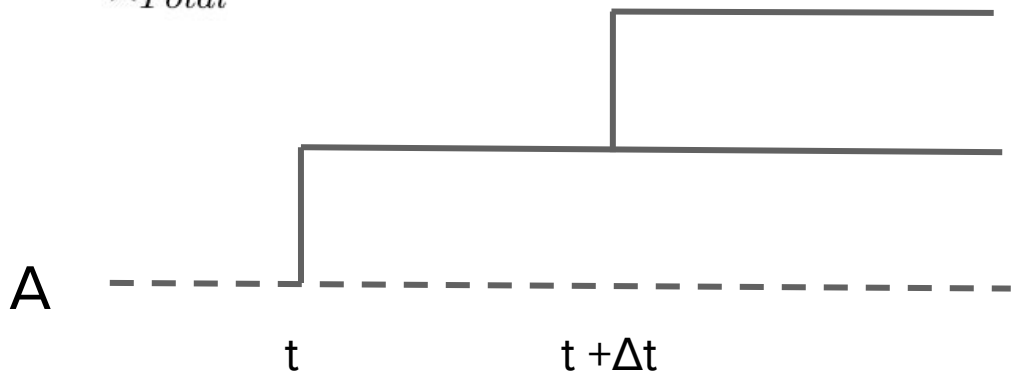
$$\lambda_{Total} = \lambda_{AT} + \lambda_{AC} + \lambda_{AG}$$



Simulating sequences on trees

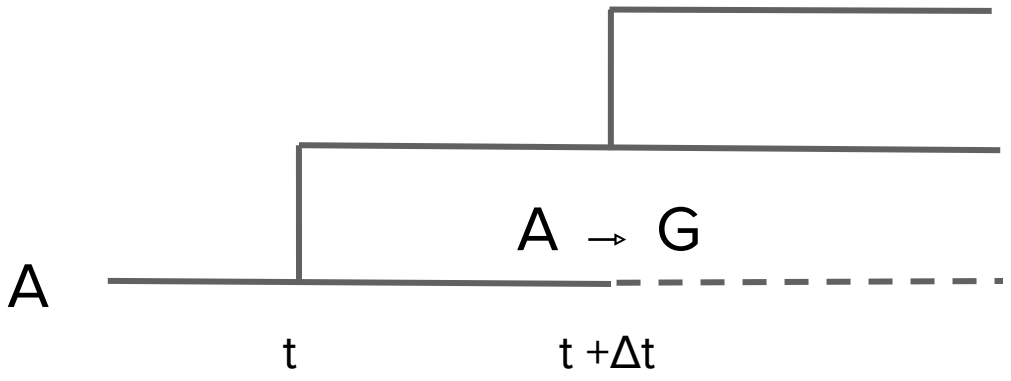
Step 2: Choose the type of substitution event the occurs at the next event time:

$$P(A \rightarrow G) = \frac{\lambda_{AG}}{\lambda_{Total}}$$



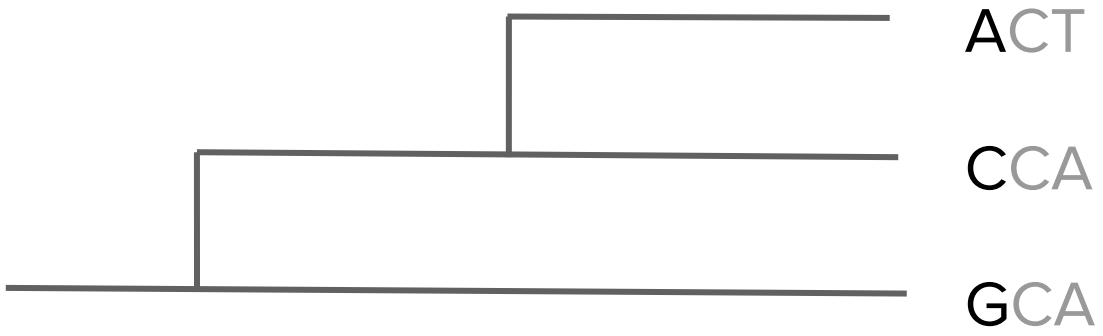
Simulating sequences on trees

Step 3: Update the time and the state of the lineage to reflect the mutation event:



Simulating sequences on trees

We can repeat this process for all lineages and all sites in order to generate mock sequence data at the tips:



Packages for simulating sequence data

Popular choices include:

- Seq-Gen https://github.com/rambaut/Seq-Gen
- Seq-Gen implementation in BEAST2

  https://www.beast2.org/2014/04/28/simulation-studies-with-beast-2.html
- Pyvolve (Spielman & Wilke, 2015) in Python
- Note: For more complicated simulations both msprime and SLiM allow for sequences to be simulated along with trees.

Why perform simulation studies?

Why do simulation studies?

Program errors: Scientific software often contains bugs since it is generally not written or extensively tested by professional software engineers

Why do simulation studies?

Program errors: Scientific software often contains bugs since it is generally not written or extensively tested by professional software engineers.

User error: It’s easy to make simple mistakes as an end-user, especially because a lot of software is both complicated and poorly documented

Why do simulation studies?

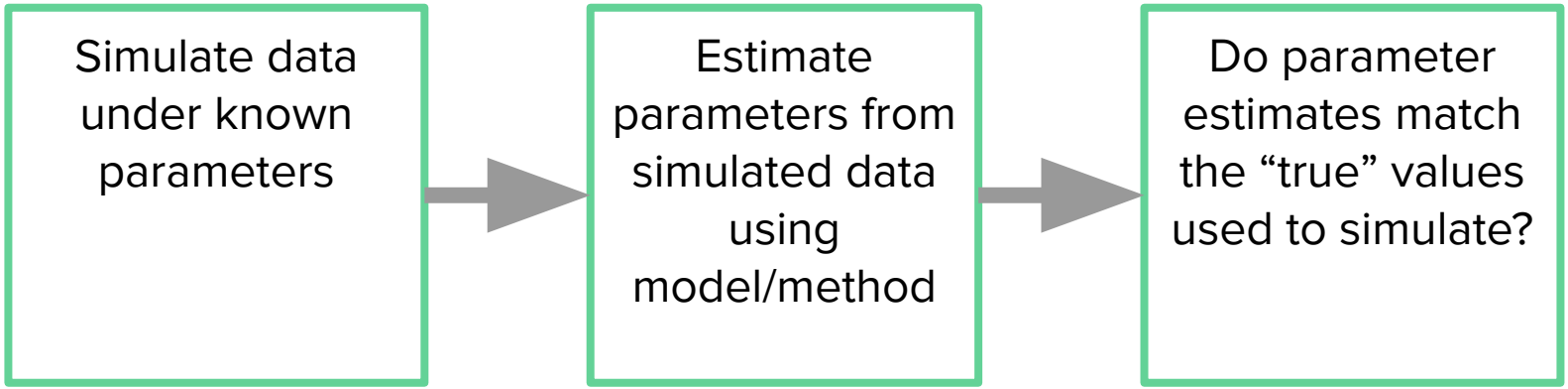
Program errors: Scientific software often contains bugs since it is generally not written or extensively tested by professional software engineers.

User error: It’s easy to make simple mistakes as an end-user, especially because a lot of software is both complicated and poorly documented

Statistical validity: Inferential methods often make many assumptions that may not be appropriate for your study system or data.

Simulation as sanity check

Can we get back out what we put in?



Why do simulation studies?

Program errors: Scientific software often contains bugs since it is generally not written or extensively tested by professional software engineers.

User error: It’s easy to make simple mistakes as an end-user, especially because a lot of software is both complicated and poorly documented

Statistical validity: Inferential methods often make many assumptions that may not be appropriate for your study system or data.

Study design: Can we learn what we want to know from realistic amounts of simulated data?

Performing simple simulation studies identifies preventable errors and makes the computational parts of science much more reproducible.