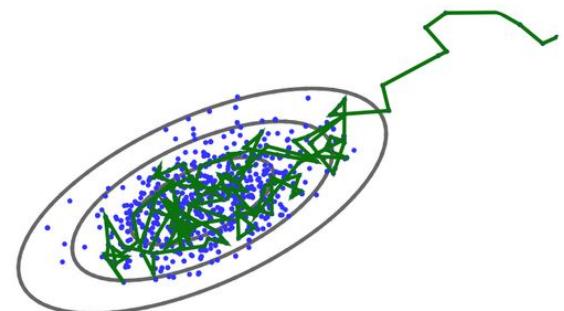
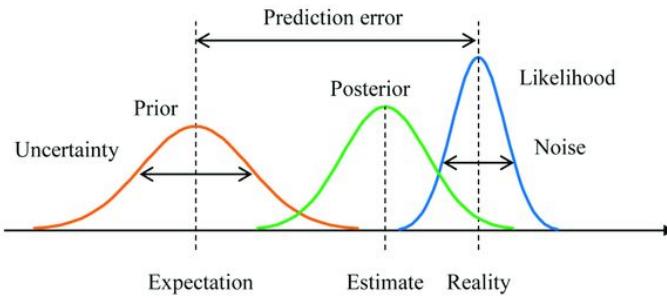


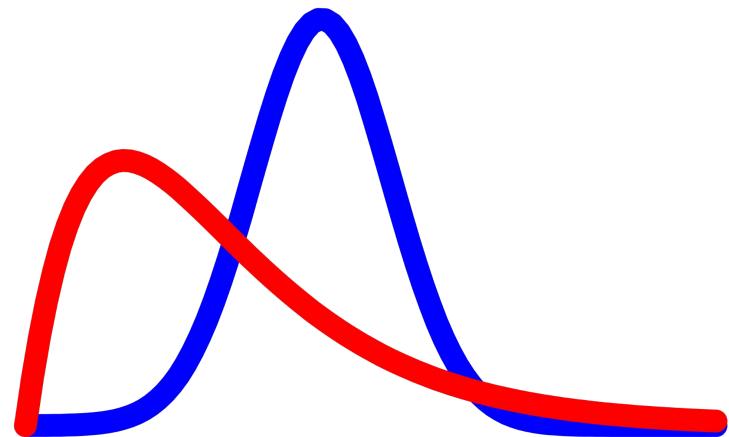
What is Chain Monte Carlo (MCMC)

Wytamma Wirth



Overview

- MCMC
- Bayes
- MCMC + Bayes
- MCMC + Phylodynamics



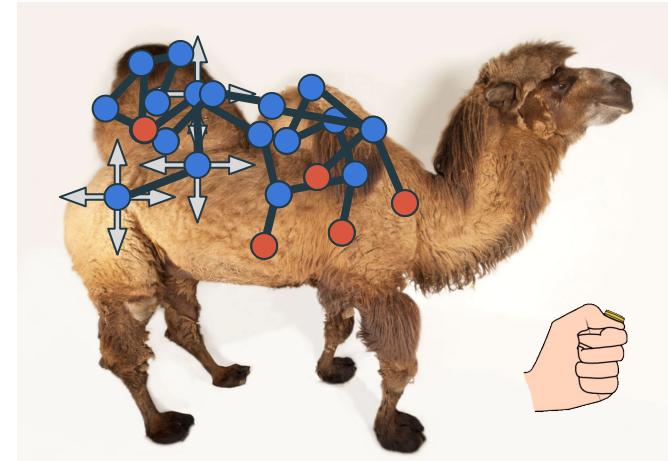
Alice the blind camel veterinarian

- Alice is a blind veterinarian investigating an outbreak of Top Hump Disease in camels.
- This disease causes lesions that have a high probability of developing at the top of the camel's humps.
- Alice cannot see the lesions.
- **How can Alice efficiently sample for the lesions without knowing exactly how they are distributed on the camel?**



Alice's solution

- Take a sample at a random location on the camel
- Move in a random direction.
- If your hand moves higher up the camel's back, sample at this new position.
- If, however, you move down the camel's back do not sample, instead flip a coin.
 - If the coin comes up heads (Alice can feel the face), accept this new lower position and take a sample,
 - if the coin is tails, move back to the previous position
- Repeat this process until you are out of sample vials (randomly moving, sampling if higher, or sampling if lower and flips a heads).



Markov Chain Monte Carlo (MCMC)

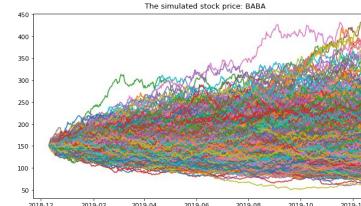
MCMC algorithms allow us to approximate a sample from an unknown distribution.

Monte Carlo simulation

Predict the probability of different outcomes when random variables are present e.g. repeated (large number) random (pseudo-random) sampling to obtain numerical results.

Markov Chain

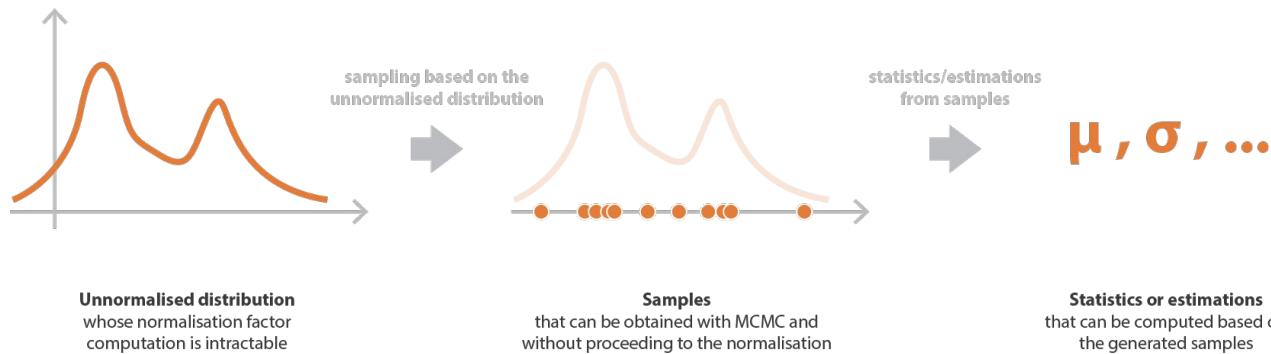
A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.

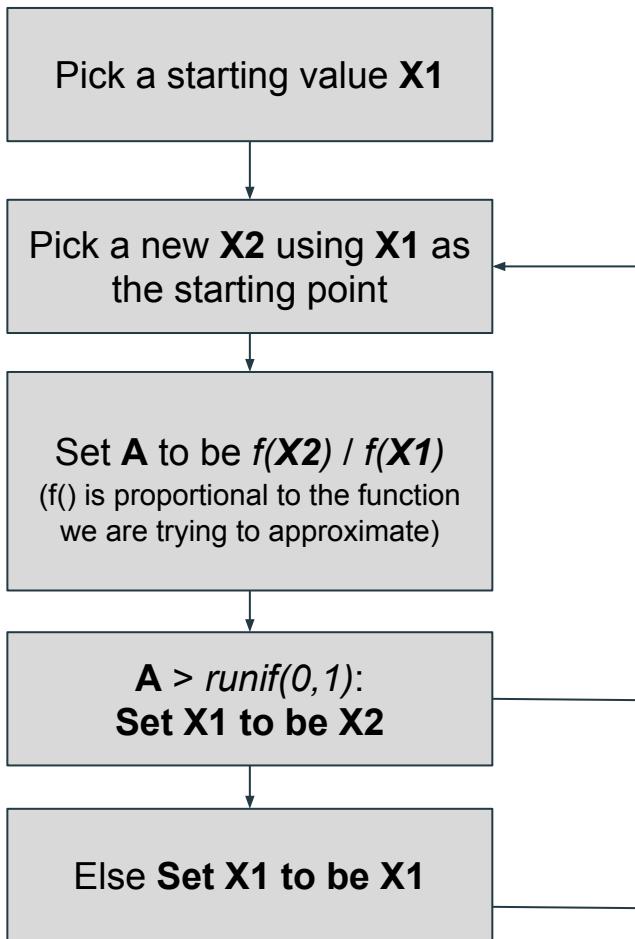


Markov Chain Monte Carlo (MCMC)

A group of sampling algorithms (most famous Metropolis–Hastings algorithm) for sampling from a probability distribution that is known up to a normalising constant.

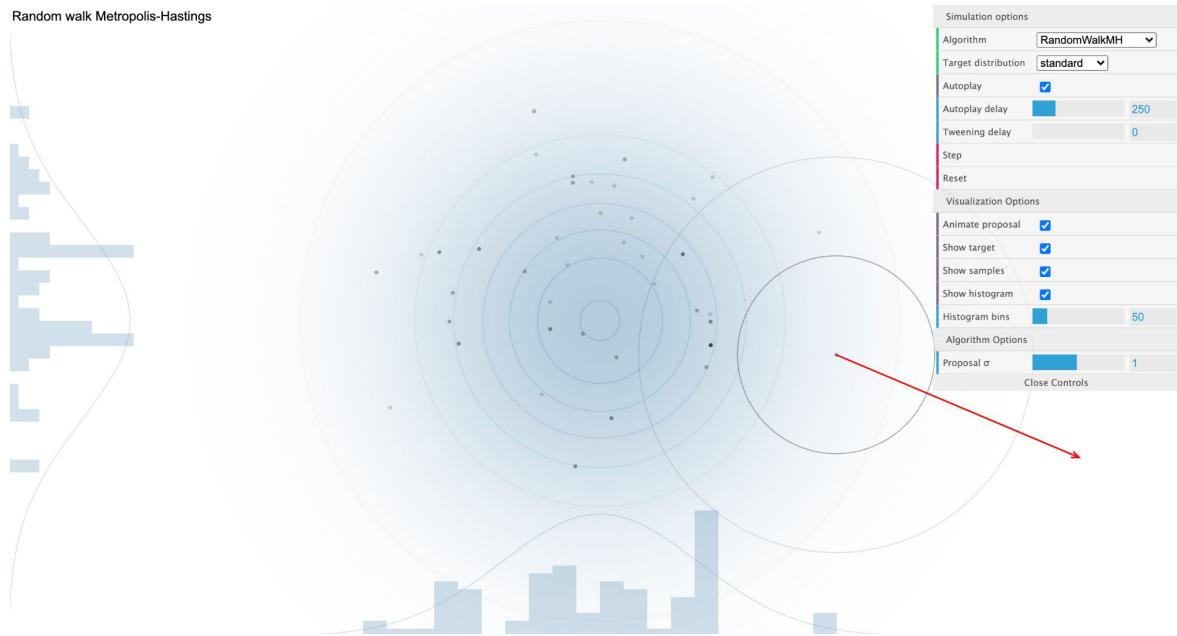
By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by recording states from the chain.





MCMC app

<https://chi-feng.github.io/mcmc-demo/app.html>



Bayesian inference

Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available.

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

LIKELIHOOD
the probability of "B" being TRUE given that "A" is TRUE

PRIOR
the probability of "A" being TRUE

POSTERIOR
the probability of "A" being TRUE given that "B" is TRUE

The probability of "B" being TRUE

That pesky evidence...

$$p(\theta|X) = \frac{p(X|\theta) p(\theta)}{p(X)}$$



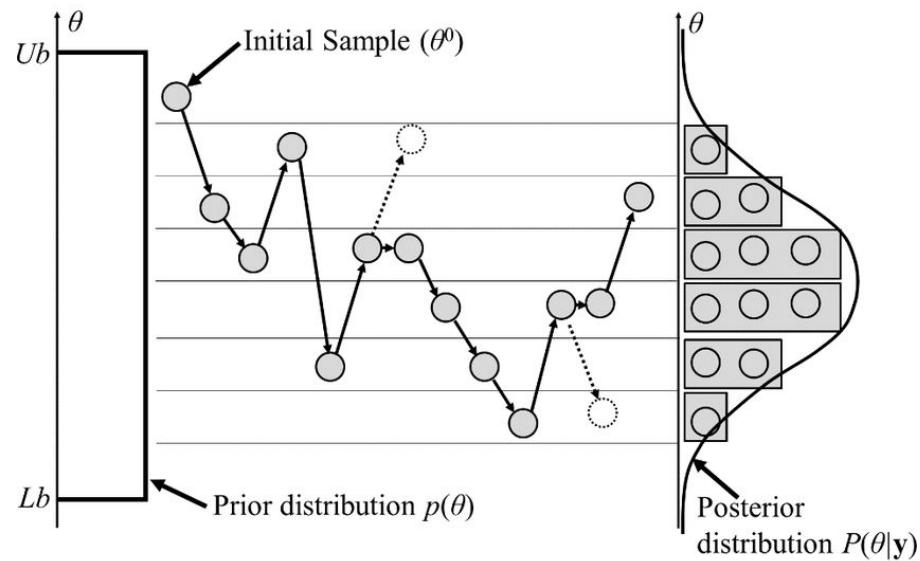
$$= \sum_T \int_{\theta} p(D | T, \theta, M) p(T, \theta | M) d\theta,$$

MCMC Magic

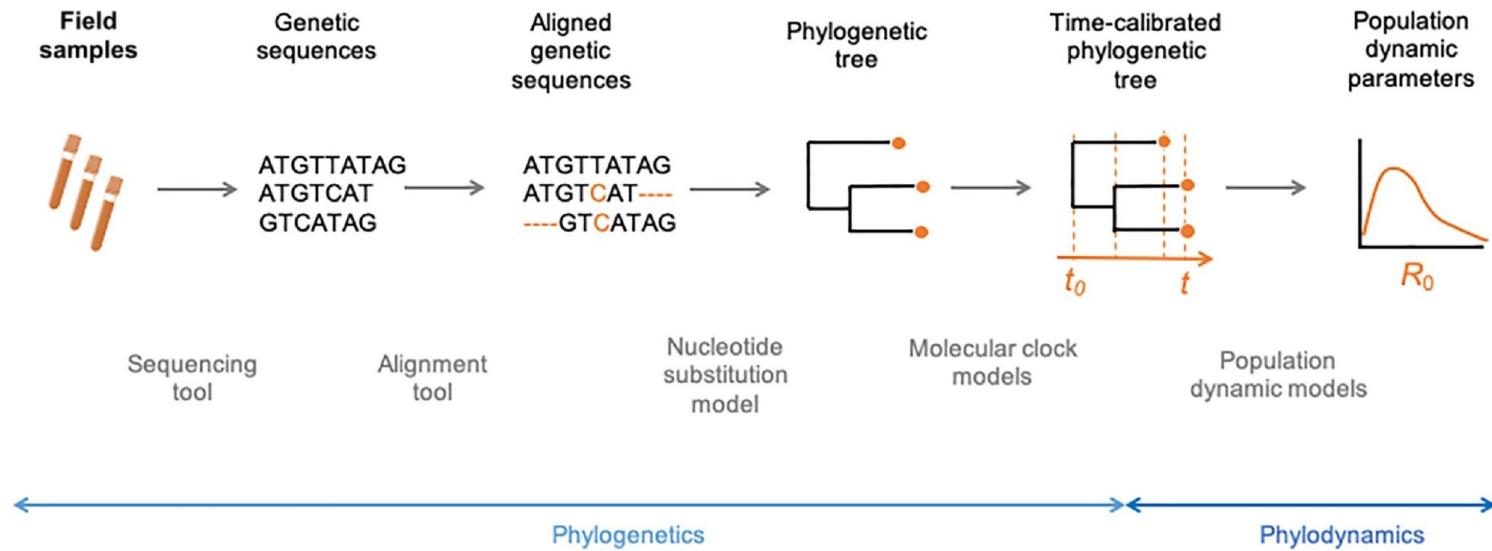
$$\frac{P(\theta_{\text{new}}|D)}{P(\theta_{\text{current}}|D)} = \frac{\frac{P(D|\theta_{\text{new}})P(\theta_{\text{new}})}{P(D)}}{\frac{P(D|\theta_{\text{current}})P(\theta_{\text{current}})}{P(D)}} = \frac{P(D|\theta_{\text{new}})P(\theta_{\text{new}})}{P(D|\theta_{\text{current}})P(\theta_{\text{current}})}$$

Using MCMC we do not have to calculate the normalising constant $P(D)$

Bayesian MCMC sampler



Phylodynamics



Phylodynamics

$$P(E \text{ } \circ\circ \text{ } \square\square \text{ } O | ACAC \dots) \underset{\text{---}}{OP} (ACAC \dots | E \text{ } \circ\circ \text{ } \square\square \text{ } O) P(E | \circ\circ) P(\circ\circ) P(\square\square) P(O)$$


genetic
sequences

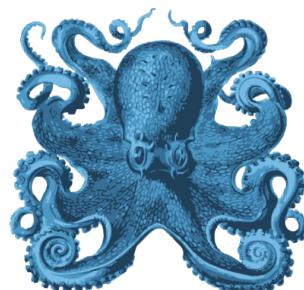

genealogy


demographic
model


substitution
model


molecular clock
model

Phylogenetics MCMC



BEAST

Bayesian Evolutionary Analysis Sampling Trees

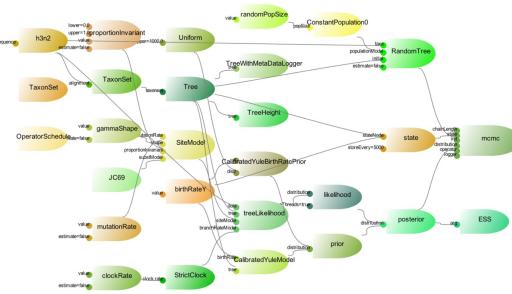


Beast2

Bayesian evolutionary analysis by sampling trees

The anatomy of a BEAST XML file

- BEAST (1 and 2) uses Extensible Markup Language (XML) files to configure analyses.
- **The BEAST XML file is the complete* description of the analysis.**
 - Data
 - Models
 - Parameters
 - Priors
 - MCMC (e.g., the chain length and operators).
- The XML file is ultimately parsed
- Important to understand the structure of this file and how to interpret its context.



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
> <beast beautitemplate='Standard' beautistatus='' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">...  
</beast>
```

BEAST v2.6.5 XML Reference

<https://www.beast2.org/xml/>

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

>   <data id="MERS_tutorial" spec="Alignment" name="alignment">...
>     </data>

>   <!-- maps -->...
>   <!-- maps -->

>   <run id="mcmc" spec="MCMC" chainLength="10000000">...
>     </run>

</beast>
```



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='1' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

    <data id="MERS_tutorial" spec="Alignment" name="alignment">...
    </data>

    <!-- maps -->
    <map name="Uniform" >beast.math.distributions.Uniform</map>
    <map name="Exponential" >beast.math.distributions.Exponential</map>
    <map name="LogNormal" >beast.math.distributions.LogNormalDistributionModel</map>
    <map name="Normal" >beast.math.distributions.Normal</map>
    <map name="Beta" >beast.math.distributions.Beta</map>
    <map name="Gamma" >beast.math.distributions.Gamma</map>
    <map name="LaplaceDistribution" >beast.math.distributions.LaplaceDistribution</map>
    <map name="prior" >beast.math.distributions.Prior</map>
    <map name="InverseGamma" >beast.math.distributions.InverseGamma</map>
    <map name="OneOnX" >beast.math.distributions.OneOnX</map>
    <!-- maps -->

    <run id="mcmc" spec="MCMC" chainLength="10000000">...
    </run>

</beast>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

    <data id="MERS_tutorial" spec="Alignment" name="alignment">...
    </data>

    <!-- maps -->...
    <!-- maps -->

    <run id="mcmc" spec="MCMC" chainLength="10000000">
        <state id="state" spec="State" storeEvery="5000">...
        </state>

        <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">...
        </init>

        <distribution id="posterior" spec="util.CompoundDistribution">...
        </distribution>

        <!-- operators -->...
        <!-- operators -->

        <!-- loggers -->...
        <!-- loggers -->

        <operatorschedule id="OperatorSchedule" spec="OperatorSchedule"/>

    </run>

</beast>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='1' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

    <data id="MERS_tutorial" spec="Alignment" name="alignment">...
    </data>

    <!-- maps -->...
    <!-- maps -->

    <run id="mcmc" spec="MCMC" chainLength="10000000">
        <state id="state" spec="State" storeEvery="5000">
            <tree id="Tree.t:MERS_tutorial" spec="beast.evolution.tree.Tree" name="stateNode">...
                </tree>
                <parameter id="clockRate.c:MERS_tutorial" spec="parameter.RealParameter" name="stateNode">6.3E-4</parameter>
                <parameter id="kappa.s:MERS_tutorial" spec="parameter.RealParameter" lower="0.0" name="stateNode">2.0</parameter>
                <parameter id="mutationRate.s:MERS_tutorial" spec="parameter.RealParameter" name="stateNode">1.0</parameter>
                <parameter id="ePopSize.t:MERS_tutorial" spec="parameter.RealParameter" name="stateNode">0.3</parameter>
                <parameter id="growthRate.t:MERS_tutorial" spec="parameter.RealParameter" name="stateNode">3.0E-4</parameter>
            </state>

            <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">
                <populationModel id="ConstantPopulation0.t:MERS_tutorial" spec="ConstantPopulation">...
                </populationModel>
            </init>

            <distribution id="posterior" spec="util.CompoundDistribution">...
            </distribution>

            <!-- operators -->...
            <!-- operators -->

            <!-- loggers -->...
            <!-- loggers -->

            <operatorSchedule id="OperatorSchedule" spec="OperatorSchedule"/>
        </run>
    </beast>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='1' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

>   <data id="MERS_tutorial" spec="Alignment" name="alignment">...
>     </data>

>   <!-- maps -->...
>   <!-- maps -->

<run id="mcmc" spec="MCMC" chainLength="10000000">
  <state id="state" spec="State" storeEvery="5000">
    <tree id="Tree.t:MERS_tutorial" spec="beast.evolution.tree.Tree" name="stateNode">
      <trait id="dateTrait:t:MERS_tutorial" spec="beast.evolution.tree.TraitSet" traitname="date" value="MERS-MN654995.1|Saudi_Arabia|Camel|2018.18=2018.18,MERS-MN655001.1|Saudi_Arabia|Camel|2018.25=2018.25,MERS-MN655006.1|Saudi_Arabia|Camel|2018.18=2018.18,MERS-MH734114.1|Kenya|Camel|2018.19=2018.19,MERS-MH734115.1|Kenya|Camel|2018.19=2018.19,MERS-NC_038294.1|United_Kingdom|Human|2012.69=2012.69,MERS-MH432120.1|Saudi_Arabia|Human|2017.16=2017.16,MERS-MH454272.1|Saudi_Arabia|Human|2016.93=2016.93,MERS-MG923466.1|Ethiopia|Camel|2017.2=2017.2,MERS-MG923467.1|Ethiopia|Camel|2017.2=2017.2,MERS-MG923468.1|Ethiopia|Camel|2017.2=2017.2,MERS-MG923469.1|Morocco|Camel|2015.17=2015.17,MERS-MG923470.1|Burkina_Faso|Camel|2015.15=2015.15,MERS-MG923471.1|Burkina_Faso|Camel|2015.21=2015.21,MERS-MG923472.1|Nigeria|Camel|2015.03=2015.03,MERS-MG923473.1|Burkina_Faso|Camel|2015.2=2015.2,MERS-MG923474.1|Nigeria|Camel|2016.06=2016.06,MERS-MG923475.1|Nigeria|Camel|2016.09=2016.09,MERS-MG923476.1|Nigeria|Camel|2016.12=2016.12,MERS-MG923477.1|Nigeria|Camel|2016.13=2016.13,MERS-MG923478.1|Nigeria|Camel|2016.09=2016.09,MERS-MG923479.1|Nigeria|Camel|2016.1=2016.1,MERS-MG923480.1|Nigeria|Camel|2016.1=2016.1,MERS-MG923481.1|Nigeria|Camel|2016.13=2016.13,MERS-KT029139.1|South_Korea|Human|2015.38=2015.38,MERS-KJ813439.1|USA|Human|2014.33=2014.33,MERS-KC164505.2|United_Kingdom|Human|2012.69=2012.69,MERS-MT226607.1|Saudi_Arabia|Camel|2017.08=2017.08,MERS-MT226606.1|Saudi_Arabia|Camel|2017.08=2017.08,MERS-MT226605.1|Saudi_Arabia|Camel|2017.08=2017.08,MERS-MT226604.1|Saudi_Arabia|Camel|2017.08=2017.08,MERS-MT226603.1|Saudi_Arabia|Camel|2017.08=2017.08,MERS-MT226601.1|Saudi_Arabia|Camel|2017.08=2017.08,MERS-MT226600.1|Saudi_Arabia|Camel|2017.08=2017.08,MERS-KU740200.1|Egypt|Camel|2014.96=2014.96,MERS-NC_019843.3||Human|2012.45=2012.45,MERS-MT387202.1|South_Korea|Human|2015.38=2015.38,MERS-MN723544.1|Saudi_Arabia|Human|2018.66=2018.66,MERS-MN723543.1|Saudi_Arabia|Human|2017.62=2017.62,MERS-MN723542.1|Saudi_Arabia|Human|2017.58=2017.58,MERS-MK280984.2|Qatar|Human|2015.38=2015.38,MERS-MN365233.1|Saudi_Arabia|Human|2019.12=2019.12,MERS-MN365232.1|Saudi_Arabia|Human|2019.07=2019.07,MERS-MN120514.1|Saudi_Arabia|Human|2019.24=2019.24,MERS-MN120513.1|Saudi_Arabia|Human|2019.23=2019.23,MERS-MK564475.1|Ethiopia|Camel|2017.64=2017.64,MERS-MK796425.1|South_Korea|Human|2015.38=2015.38,MERS-MK564474.1|Ethiopia|Camel|2017.64=2017.64,MERS-MH259486.1|Saudi_Arabia|Camel|2014.53=2014.53,MERS-MH259485.1|Saudi_Arabia|Camel|2014.53=2014.53,MERS-MH029552.1|Saudi_Arabia|Human|2015.65=2015.65,MERS-MK483839.1|Saudi_Arabia|Human|2018.62=2018.62,MERS-MK462256.1|Saudi_Arabia|Human|2018.7=2018.7,MERS-MK462255.1|Saudi_Arabia|Human|2018.7=2018.7,MERS-MK462254.1|Saudi_Arabia|Human|2018.66=2018.66,MERS-MK462253.1|Saudi_Arabia|Human|2018.66=2018.66,MERS-MK462252.1|Saudi_Arabia|Human|2018.59=2018.59,MERS-MK462251.1|Saudi_Arabia|Human|2018.54=2018.54,MERS-MK462250.1|Saudi_Arabia|Human|2018.53=2018.53,MERS-MK462249.1|Saudi_Arabia|Human|2018.52=2018.52,MERS-MK462248.1|Saudi_Arabia|Human|2018.45=2018.45,MERS-MK462247.1|Saudi_Arabia|Human|2018.43=2018.43,MERS-MK462246.1|Saudi_Arabia|Human|2017.62=2017.62,MERS-MK462245.1|Saudi_Arabia|Human|2017.62=2017.62,MERS-MK462244.1|Saudi_Arabia|Human|2017.6=2017.6,MERS-MK462243.1|Saudi_Arabia|Human|2017.58=2017.58,MERS-MK129253.1|South_Korea|Human|2018.8=2018.8,MERS-MG757605.1|Saudi_Arabia|Human|2016.28=2016.28,MERS-MG757604.1|Saudi_Arabia|Human|2016.08=2016.08,MERS-MG757603.1|Saudi_Arabia|Human|2015.71=2015.71,MERS-MG757602.1|Saudi_Arabia|Human|2015.67=2015.67,MERS-MG757601.1|Saudi_Arabia|Human|2015.67=2015.67,MERS-MG757600.1|Saudi_Arabia|Human|2015.67=2015.67,MERS-MG757599.1|Saudi_Arabia|Human|2015.65=2015.65,MERS-MG757598.1|Saudi_Arabia|Human|2015.65=2015.65,MERS-MG757597.1|Saudi_Arabia|Human|2015.64=2015.64,MERS-MG757596.1|Saudi_Arabia|Human|2015.64=2015.64,
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='1' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

    <data id="MERS_tutorial" spec="Alignment" name="alignment">...
    </data>

    <!-- maps -->...
    <!-- maps -->

    <run id="mcmc" spec="MCMC" chainLength="10000000">
        <state id="state" spec="State" storeEvery="5000">
            <tree id="Tree.t:MERS_tutorial" spec="beast.evolution.tree.Tree" name="stateNode">...
                </tree>
                <parameter id="clockRate.c:MERS_tutorial" spec="parameter.RealParameter" name="stateNode">6.3E-4</parameter>
                <parameter id="kappa.s:MERS_tutorial" spec="parameter.RealParameter" lower="0.0" name="stateNode">2.0</parameter>
                <parameter id="mutationRate.s:MERS_tutorial" spec="parameter.RealParameter" name="stateNode">1.0</parameter>
                <parameter id="ePopSize.t:MERS_tutorial" spec="parameter.RealParameter" name="stateNode">0.3</parameter>
                <parameter id="growthRate.t:MERS_tutorial" spec="parameter.RealParameter" name="stateNode">3.0E-4</parameter>
            </state>

            <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">
                <populationModel id="ConstantPopulation0.t:MERS_tutorial" spec="ConstantPopulation">...
                </populationModel>
            </init>

            <distribution id="posterior" spec="util.CompoundDistribution">...
            </distribution>

            <!-- operators -->...
            <!-- operators -->

            <!-- loggers -->...
            <!-- loggers -->

            <operatorSchedule id="OperatorSchedule" spec="OperatorSchedule"/>
        </run>
    </beast>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

    <data id="MERS_tutorial" spec="Alignment" name="alignment">...
    </data>

    <!-- maps -->...
    <!-- maps -->

    <run id="mcmc" spec="MCMC" chainLength="10000000">
        <state id="state" spec="State" storeEvery="5000">...
        </state>

        <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">...
        </init>

        <distribution id="posterior" spec="util.CompoundDistribution">
            <distribution id="prior" spec="util.CompoundDistribution">...
            </distribution>
            <distribution id="likelihood" spec="util.CompoundDistribution" useThreads="true">...
            </distribution>
        </distribution>

        <!-- operators -->...
        <!-- operators -->

        <!-- loggers -->...
        <!-- loggers -->

        <operatorschedule id="OperatorSchedule" spec="OperatorSchedule"/>

    </run>
</beast>
```


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

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='1' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">
```

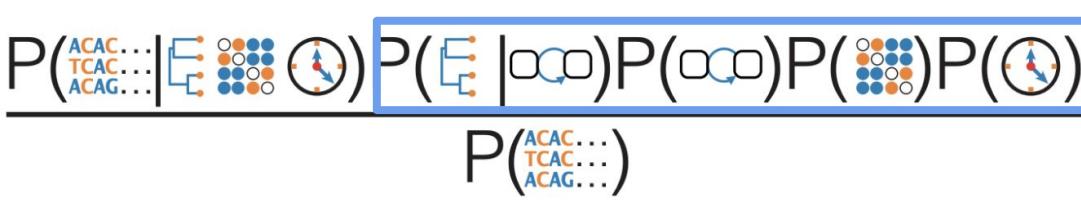
```
>   <data id="MERS_tutorial" spec="Alignment" name="alignment">...
</data>
```

```
>   <!-- maps -->...
<!-- maps -->
```

```
>   <run id="mcmc" spec="MCMC" chainLength="10000000">
>     <state id="state" spec="State" storeEvery="5000">...
</state>
```

```
>   <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">...
</init>
```

```
<distribution id="posterior" spec="util.CompoundDistribution">
  <distribution id="prior" spec="util.CompoundDistribution">
    <distribution id="CoalescentExponential.t:MERS_tutorial" spec="Coalescent">
      <populationModel id="ExponentialGrowth.t:MERS_tutorial" spec="ExponentialGrowth" growthRate="@growthRate.t:MERS_tutorial"
        popSize="@ePopSize.t:MERS_tutorial"/>
      <treeIntervals id="TreeIntervals.t:MERS_tutorial" spec="TreeIntervals" tree="@Tree.t:MERS_tutorial"/>
    </distribution>
    <prior id="ClockPrior.c:MERS_tutorial" name="distribution" x="@clockRate.c:MERS_tutorial">
      <Gamma id="Gamma.0" name="distr">
        <parameter id="RealParameter.6" spec="parameter.RealParameter" estimate="false" name="alpha">2.0</parameter>
        <parameter id="RealParameter.7" spec="parameter.RealParameter" estimate="false" name="beta">2.0</parameter>
      </Gamma>
    </prior>
    <prior id="ePopSizePrior.t:MERS_tutorial" name="distribution" x="@ePopSize.t:MERS_tutorial">
      <Exponential id="Exp" name="distr">
        <parameter id="RealParameter.8" spec="parameter.RealParameter" estimate="false" name="rate">0.0001</parameter>
      </Exponential>
    </prior>
    <prior id="GrowthRatePrior.t:MERS_tutorial" name="distribution" x="@growthRate.t:MERS_tutorial">
      <LaplaceDistribution id="Lap" name="distr">
        <parameter id="RealParameter.9" spec="parameter.RealParameter" estimate="false" name="mu">0.0001</parameter>
        <parameter id="RealParameter.10" spec="parameter.RealParameter" estimate="false" name="sigma">0.0001</parameter>
      </LaplaceDistribution>
    </prior>
    <prior id="KappaPrior.s:MERS_tutorial" name="distribution" x="@kappa.s:MERS_tutorial">
```



```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='1' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

>   <data id="MERS_tutorial" spec="Alignment" name="alignment">...
>     </data>

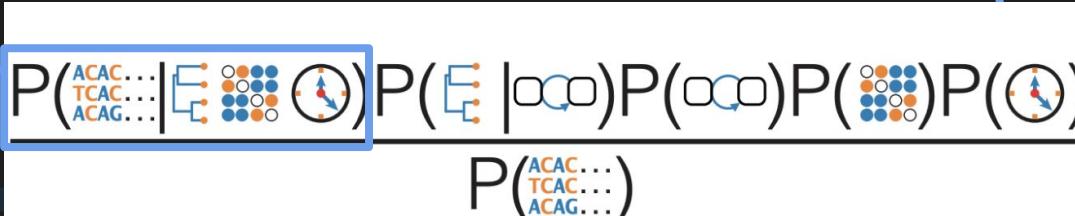
>   <!-- maps -->...
>   <!-- maps -->

>   <run id="mcmc" spec="MCMC" chainLength="10000000">
>     <state id="state" spec="State" storeEvery="5000">...
>       </state>

>     <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">...
>       </init>

>     <distribution id="posterior" spec="util.CompoundDistribution">
>       <distribution id="prior" spec="util.CompoundDistribution">...
>         </distribution>
>         <distribution id="likelihood" spec="util.CompoundDistribution" useThreads="true">
>           <distribution id="treeLikelihood.MERS_tutorial" spec="ThreadedTreeLikelihood" data="@MERS_tutorial" tree="@Tree.t:MERS_tutorial">
>             <siteModel id="SiteModel.s:MERS_tutorial" spec="SiteModel" mutationRate="@mutationRate.s:MERS_tutorial">
>               <parameter id="gammaShape.s:MERS_tutorial" spec="parameter.RealParameter" estimate="false" name="shape">1.0</parameter>
>               <parameter id="proportionInvariant.s:MERS_tutorial" spec="parameter.RealParameter" estimate="false" lower="0.0" name="proportionInvariant" upper="1.0">0.0</parameter>
>               <substModel id="hky.s:MERS_tutorial" spec="HKY" kappa="@kappa.s:MERS_tutorial">
>                 <frequencies id="empiricalFreqs.s:MERS_tutorial" spec="Frequencies" data="@MERS_tutorial"/>
>               </substModel>
>             </siteModel>
>             <branchRateModel id="StrictClock.c:MERS_tutorial" spec="beast.evolution.branchratemodel.StrictClockModel" clock.rate="@clockRate.c:MERS_tutorial"/>
>           </distribution>
>         </distribution>
>       </distribution>
>     </distribution>
>     <!-- operators -->...
>     <!-- operators -->
>     <!-- loggers -->...
>     <!-- loggers -->

```



$$P(\text{ACAC...} | \text{E} \circlearrowleft \text{TCAC...} \circlearrowright \text{ACAG...}) = P(\text{E} | \text{TCAC...}) P(\text{TCAC...} | \text{ACAG...}) P(\text{ACAG...})$$

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">
```

```
>   <data id="MERS_tutorial" spec="Alignment" name="alignment">...
>     </data>
```

```
>   <!-- maps -->...
>   <!-- maps -->
```

```
>   <run id="mcmc" spec="MCMC" chainLength="10000000">
>     <state id="state" spec="State" storeEvery="5000">...
>       </state>
```

```
>   <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">...
>     </init>
```

```
>   <distribution id="posterior" spec="util.CompoundDistribution">...
>     </distribution>
```

```
>     <!-- operators -->
>       <operator id="StrictClockRateScaler.c:MERS_tutorial" spec="ScaleOperator" parameter="@clockRate.c:MERS_tutorial" weight="3.0"/>
>       <operator id="strictClockUpDownOperator.c:MERS_tutorial" spec="UpDownOperator" scaleFactor="0.75" weight="3.0">...
>         </operator>
```

```
>       <operator id="KappaScaler.s:MERS_tutorial" spec="ScaleOperator" parameter="@kappa.s:MERS_tutorial" scaleFactor="0.5" weight="0.1"/>
>       <operator id="FixMeanMutationRatesOperator" spec="DeltaExchangeOperator" delta="0.75" weight="2.0">...
>         </operator>
```

```
>       <operator id="CoalescentExponentialTreeScaler.t:MERS_tutorial" spec="ScaleOperator" scaleFactor="0.5" tree="@Tree.t:MERS_tutorial" weight="3.0"/>
>       <operator id="CoalescentExponentialTreeRootScaler.t:MERS_tutorial" spec="ScaleOperator" rootOnly="true" scaleFactor="0.5" tree="@Tree.t:MERS_tutorial" weight="3.0"/>
```

```
>       <operator id="CoalescentExponentialUniformOperator.t:MERS_tutorial" spec="Uniform" tree="@Tree.t:MERS_tutorial" weight="30.0"/>
>       <operator id="CoalescentExponentialSubtreeSlide.t:MERS_tutorial" spec="SubtreeSlide" tree="@Tree.t:MERS_tutorial" weight="15.0"/>
```

```
>       <operator id="CoalescentExponentialNarrow.t:MERS_tutorial" spec="Exchange" tree="@Tree.t:MERS_tutorial" weight="15.0"/>
>       <operator id="CoalescentExponentialWide.t:MERS_tutorial" spec="Exchange" isNarrow="false" tree="@Tree.t:MERS_tutorial" weight="3.0"/>
```

```
>       <operator id="CoalescentExponentialWilsonBalding.t:MERS_tutorial" spec="WilsonBalding" tree="@Tree.t:MERS_tutorial" weight="3.0"/>
>       <operator id="ePopSizeScaler.t:MERS_tutorial" spec="ScaleOperator" parameter="@ePopSize.t:MERS_tutorial" weight="3.0"/>
```

```
>       <operator id="GrowthRateRandomWalk.t:MERS_tutorial" spec="RealRandomWalkOperator" parameter="@growthRate.t:MERS_tutorial" weight="3.0" windowSize="1.0"/>
>     <!-- operators -->
```

```
>     <!-- loggers -->...
>     <!-- loggers -->
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

    <data id="MERS_tutorial" spec="Alignment" name="alignment">...
    </data>

    <!-- maps -->...
    <!-- maps -->

    <run id="mcmc" spec="MCMC" chainLength="10000000">
        <state id="state" spec="State" storeEvery="5000">...
        </state>

        <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">...
        </init>

        <distribution id="posterior" spec="util.CompoundDistribution">...
        </distribution>

        <!-- operators -->...
        <!-- operators -->

        <!-- loggers -->
        <logger id="tracelog" spec="Logger" fileName="MERS_tutorial.log" logEvery="1000" model="@posterior" sanitiseHeaders="true" sort="smart">...
        </logger>

        <logger id="screenlog" spec="Logger" logEvery="10000">...
        </logger>

        <logger id="treelog.t:MERS_tutorial" spec="Logger" fileName="$(tree).trees" logEvery="1000" mode="tree">...
        </logger>
        <!-- loggers -->

        <operatorschedule id="OperatorSchedule" spec="OperatorSchedule"/>

    </run>

</beast>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">
```

```
>   <data id="MERS_tutorial" spec="Alignment" name="alignment">...
</data>
```

```
>   <!-- maps -->...
<!-- maps -->
```

```
>   <run id="mcmc" spec="MCMC" chainLength="10000000">
>     <state id="state" spec="State" storeEvery="5000">...
</state>
```

```
>     <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">...
</init>
```

```
>     <distribution id="posterior" spec="util.CompoundDistribution">...
</distribution>
```

```
>   <!-- operators -->...
<!-- operators -->
```

```
<!-- loggers -->
  <logger id="tracelog" spec="Logger" fileName="MERS_tutorial.log" logEvery="1000" mode="@posterior" sanitiseHeaders="true" sort="smart">
    <log idref="posterior"/>
    <log idref="likelihood"/>
    <log idref="prior"/>
    <log idref="treeLikelihood.MERS_tutorial"/>
    <log idref="TreeHeight.t:MERS_tutorial" spec="beast.evolution.tree.TreeHeightLogger" tree="@Tree.t:MERS_tutorial"/>
    <log idref="clockRate.c:MERS_tutorial"/>
    <log idref="kappa.s:MERS_tutorial"/>
    <log idref="mutationRate.s:MERS_tutorial"/>
    <log idref="CoalescentExponential.t:MERS_tutorial"/>
    <log idref="ePopSize.t:MERS_tutorial"/>
    <log idref="growthRate.t:MERS_tutorial"/>
  </logger>
```

```
>   <logger id="screenlog" spec="Logger" logEvery="10000">...
</logger>
```

```
>   <logger id="treelog.t:MERS_tutorial" spec="Logger" fileName="$(tree).trees" logEvery="1000" mode="tree">...
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<beast beautitemplate='Standard' beautistatus='1' namespace="beast.core:beast.evolution.alignment:beast.evolution.tree.coalescent:beast.core.util:beast.evolution.nuc:beast.evolution.operators:beast.evolution.sitemodel:beast.evolution.substitutionmodel:beast.evolution.likelihood" required="" version="2.6">

    <data id="MERS_tutorial" spec="Alignment" name="alignment">...
        </data>

    <!-- maps -->...
    <!-- maps -->

    <run id="mcmc" spec="MCMC" chainLength="10000000">
        <state id="state" spec="State" storeEvery="5000">...
            </state>

        <init id="RandomTree.t:MERS_tutorial" spec="beast.evolution.tree.RandomTree" estimate="false" initial="@Tree.t:MERS_tutorial" taxa="@MERS_tutorial">...
            </init>

        <distribution id="posterior" spec="util.CompoundDistribution">...
            </distribution>

        <!-- operators -->...
        <!-- operators -->

        <!-- loggers -->...
        <!-- loggers -->

        <operatorschedule id="OperatorSchedule" spec="OperatorSchedule"/>

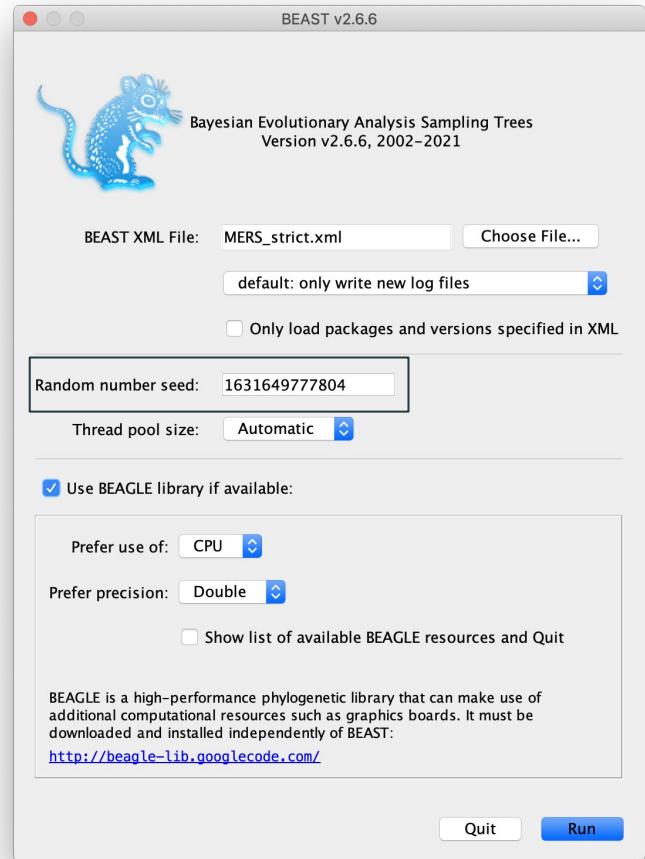
    </run>
</beast>
```

BEAST v2.6.5 XML Reference

<https://www.beast2.org/xml/>

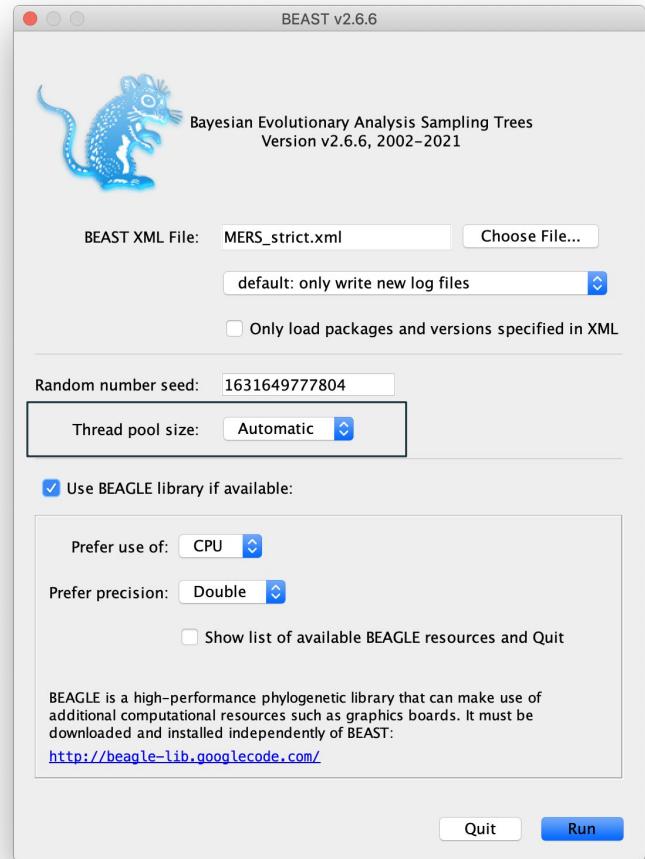
Running a MCMC in BEAST

- Most important BEAST run-time setting (for reproducibility)
- Used to initialize the pseudo-random number generator
- Because MCMC is a stochastic process replication of a run requires the same seed.
- **Report your seed when sharing results, to ensure reproducibility.**



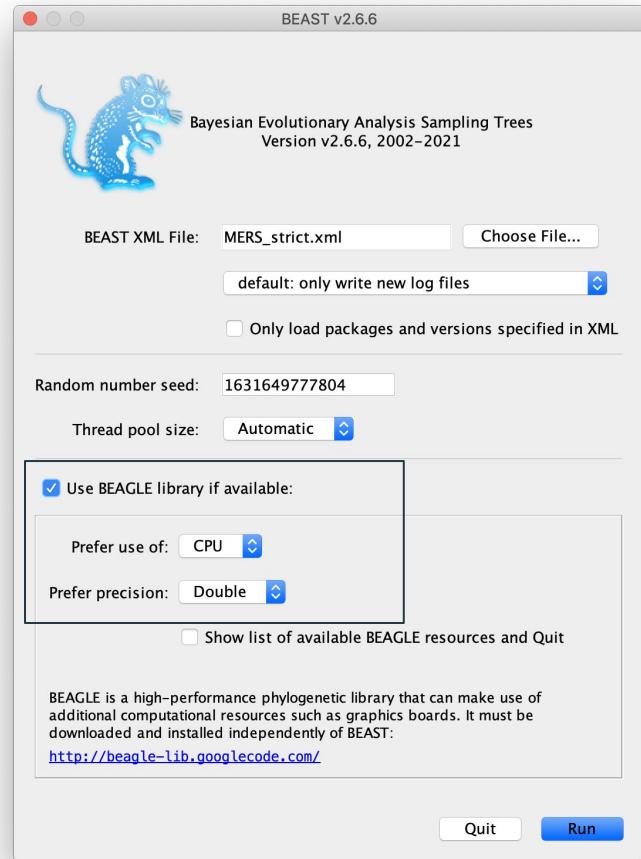
Running a MCMC in BEAST

- Generally, the most time-consuming aspect of the BEAST MCMC algorithm is calculating the tree likelihood.
- The **ThreadedTreelikelihood** package in BEAST will split the tree likelihood calculation into equal parts based on the number of threads available and compute them in parallel.



Running a MCMC in BEAST

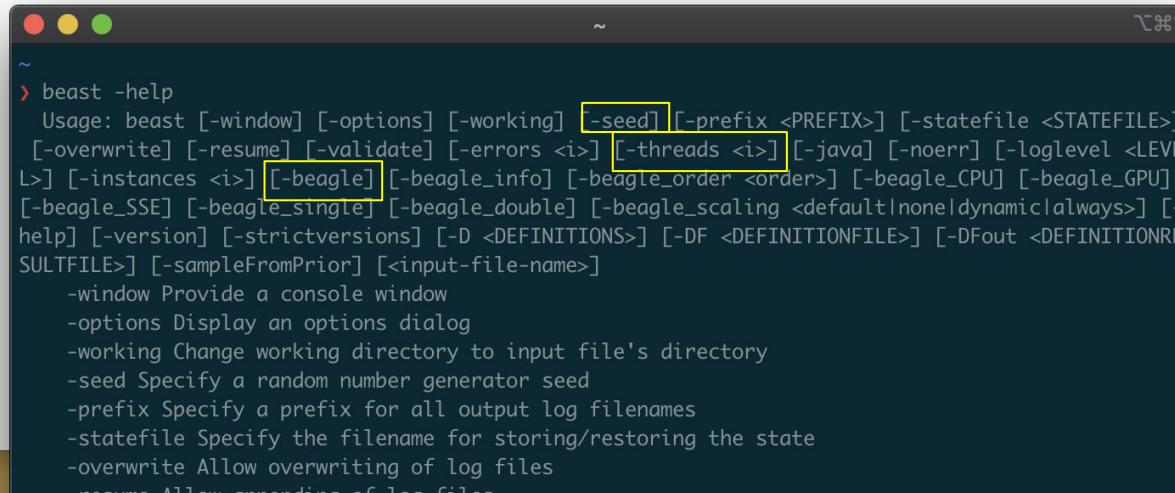
- BEAGLE is a high-performance library for performing fast tree likelihood calculations and enables BEAST to take advantage of GPUs for computation.
- Using BEAGLE is standard practice for BEAST analysis and can dramatically reduce the time required for a run.
- A GPU is not required to use BEAGLE and users can see up to 2x improvements by using BEAGLE on a CPU. Instructions for using and installing BEAGLE can be found at <https://github.com/beagle-dev/beagle-lib>.



Running BEAST from the command line

BEAST command line interface has several features that are not available from the graphical interface.

- Resuming runs
- Changing parameters at run time (e.g., seed, treads, setting the chain length)
- Modifying the output structure
- etc.



A screenshot of a terminal window on a Mac OS X system. The window title is 'Terminal'. The command entered is 'beast -help'. The output shows the usage of the 'beast' command with various options. Several options are highlighted with yellow boxes: '-seed', '-prefix <PREFIX>', '-threads <i>', '-java', '-loglevel <LEVEL>', '-instances <i>', '-beagle', and '-beagle_info'. Below the usage information, detailed descriptions are provided for each option.

```
~ > beast -help
Usage: beast [-window] [-options] [-working] [-seed] [-prefix <PREFIX>] [-statefile <STATEFILE>]
[-overwrite] [-resume] [-validate] [-errors <i>] [-threads <i>] [-java] [-noerr] [-loglevel <LEVEL>]
[-instances <i>] [-beagle] [-beagle_info] [-beagle_order <order>] [-beagle_CPU] [-beagle_GPU]
[-beagle_SSE] [-beagle_single] [-beagle_double] [-beagle_scaling <default|none|dynamic|always>]
[-help] [-version] [-strictversions] [-D <DEFINITIONS>] [-DF <DEFINITIONFILE>] [-DFout <DEFINITIONFILE>]
[-sampleFromPrior] [<input-file-name>]
    -window Provide a console window
    -options Display an options dialog
    -working Change working directory to input file's directory
    -seed Specify a random number generator seed
    -prefix Specify a prefix for all output log filenames
    -statefile Specify the filename for storing/restoring the state
    -overwrite Allow overwriting of log files
    -resume Allow resuming of log files
```

Dynamic (runtime) parameters

It can be useful to set some BEAST options at run-time without have to produce multiple XML files (during testing or simulation studies).

The BEAST command line option `‐D` is useful in this scenario. For example, if you want to change in you want to be able to set the chain length at run-time you can change the mcmc tag e.g. `<mcmc chainLength="X" ...>` to the format `<mcmc chainLength="{chainLength}" ...>` then specify the desired chain length then you run the beast command e.g., `beast –D "chainLength=10000" BEAST.xml`.

Dynamic BEAST

pypi v1.3.0  test passing  codecov 91%

This command line tool can be used to create a dynamic version of BEAST2 XML files. This dynamic XML file can be used to set BEAST parameters at runtime, which can be useful for testing different configurations or quickly modifying parameters without having to edit the XML file.



Beastiary

<https://beastiary.wytamma.com>



Traces



hcv_coal.log

tests/data/hcv_coal.log

3000000 1001 4

Burn-in 10%

| | | |
|-------------------------------------|-----------------|-----|
| <input type="checkbox"/> | freqParameter.2 | 56 |
| <input type="checkbox"/> | freqParameter.3 | 59 |
| <input type="checkbox"/> | freqParameter.4 | 85 |
| <input checked="" type="checkbox"/> | rateAC | 160 |
| <input type="checkbox"/> | rateAG | 55 |
| <input checked="" type="checkbox"/> | rateAT | 89 |
| <input checked="" type="checkbox"/> | rateCG | 132 |
| <input checked="" type="checkbox"/> | rateGT | 115 |
| <input type="checkbox"/> | gammaShape | 315 |
| <input type="checkbox"/> | BayesianSkyline | 85 |
| <input type="checkbox"/> | bPopSizes.1 | 456 |
| <input type="checkbox"/> | bPopSizes.2 | 91 |

TRACE

VIOLIN

HISTOGRAM

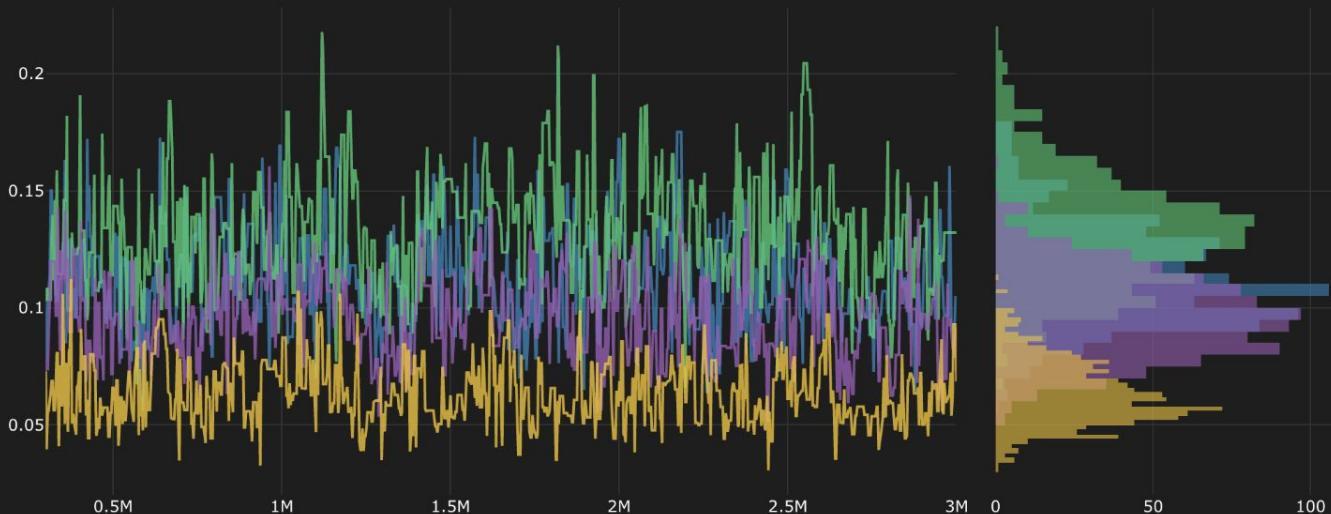
PARALLEL

JOINT

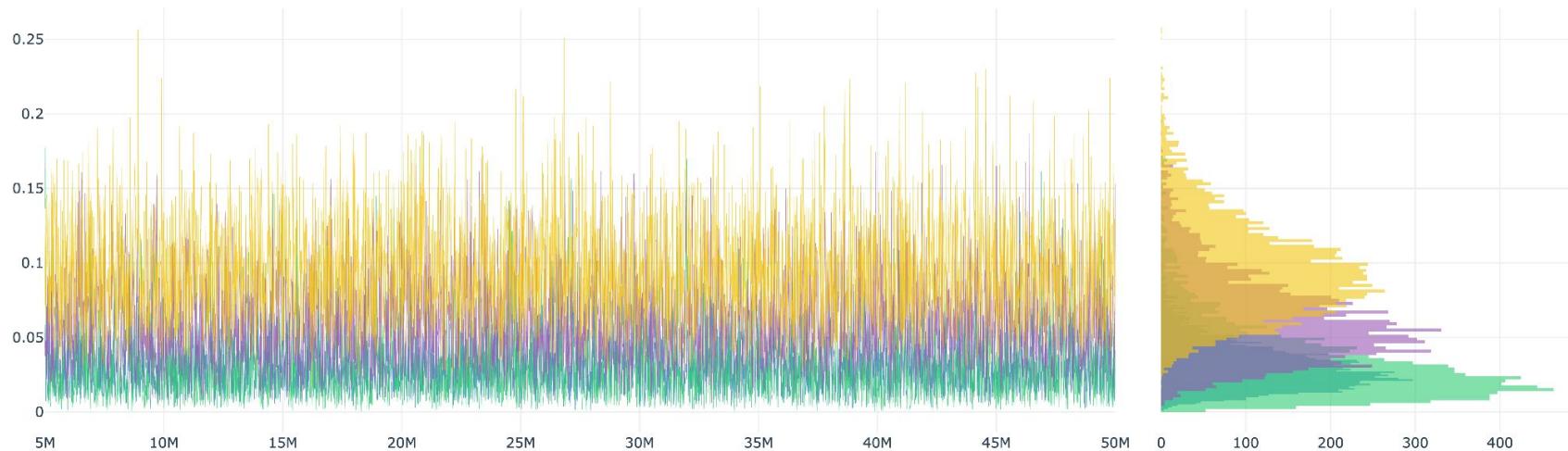
ESS

ESTIMATES

■ rateAC ■ rateAT ■ rateGT ■ rateCG



Assessing results

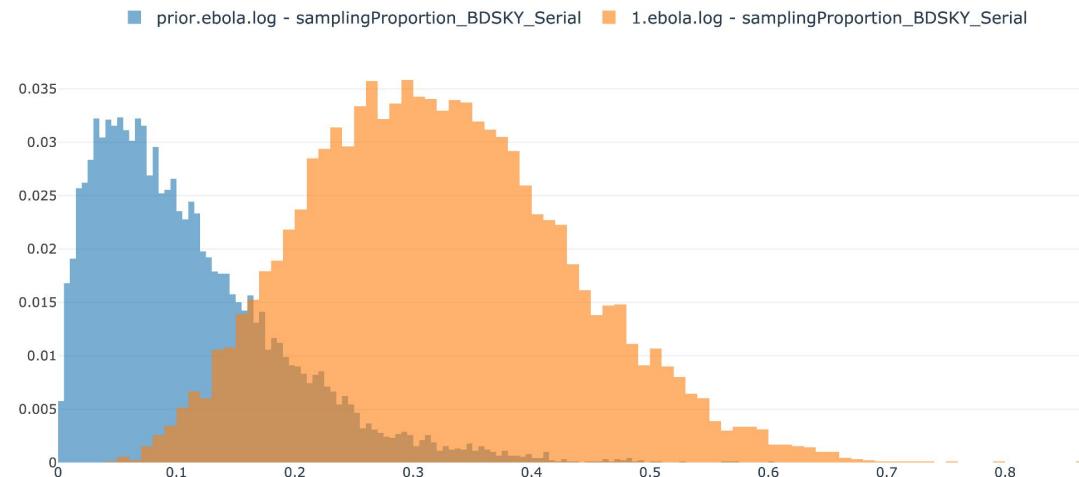


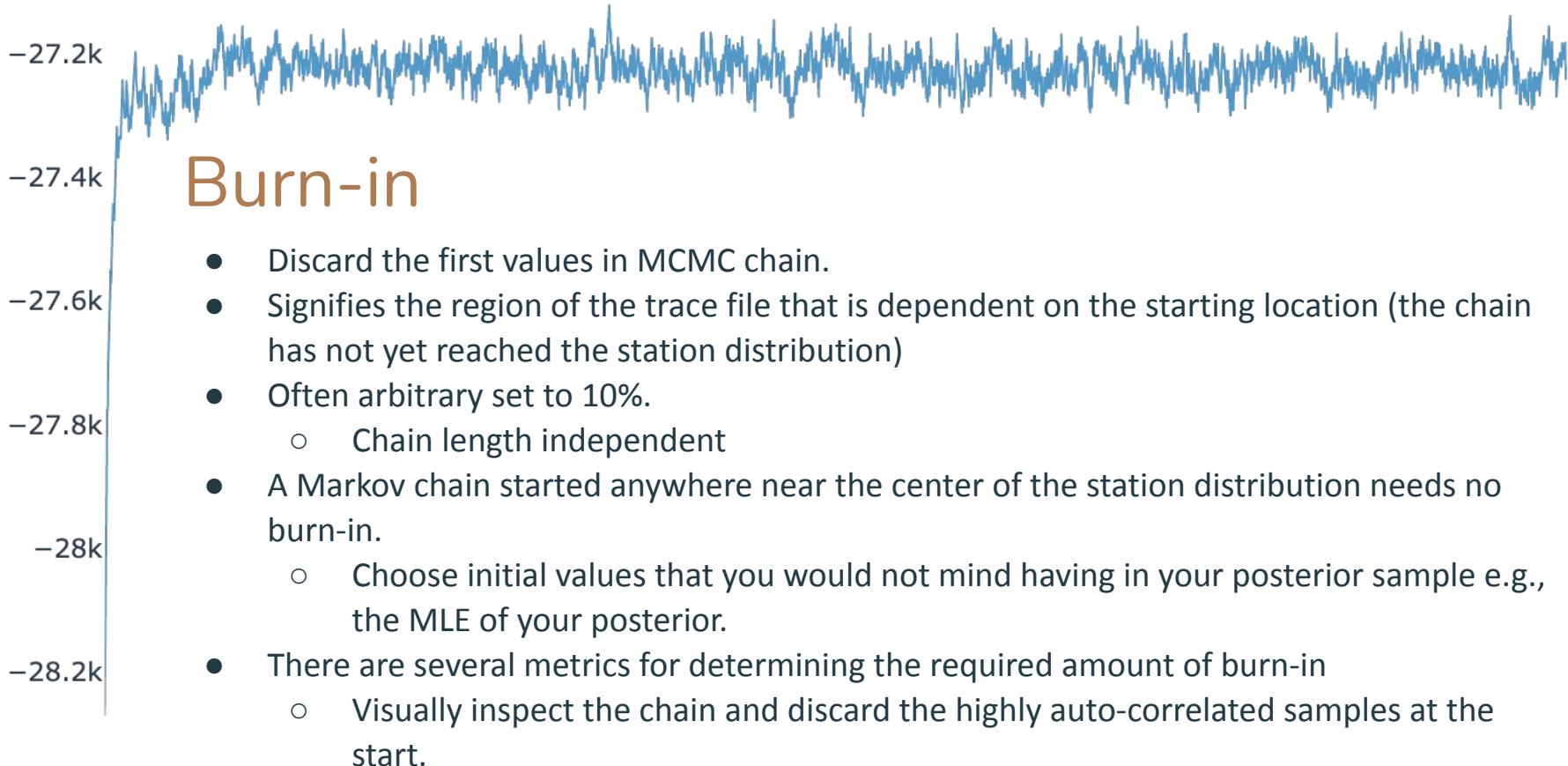
Prior predictive checking

Priors may differ from what has actually been specified by the user.

Use our MCMC to sample directly from our prior distributions to ensure that they reflect our knowledge.

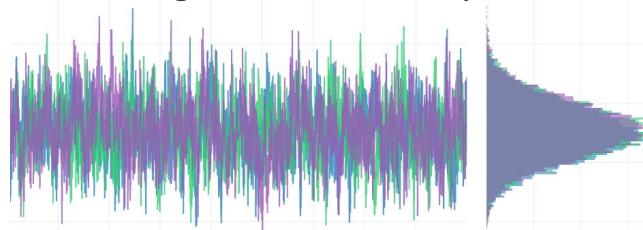
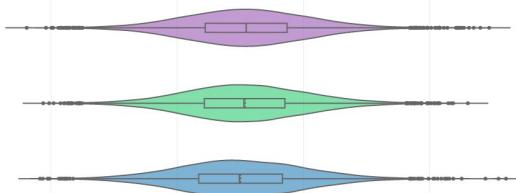
Useful for examining how much we have learnt from our data.





Convergence

- A trace is said to have converged when samples from its stationary distribution approximate that of the target distribution.
- A chain that has no obvious trends or change in spread in state (hairy caterpillar).
 - Having no trends or changes in spread does not guarantee that the chain is converged.
 - A typical way to assess the trends and spread in state is to visualize the samples as trace plot.
- How can we assess if a MCMC algorithm has converged?
- Multiple independent chains and compare their posterior distributions.
 - Replicate chains that fail to produce the same parameter estimates are indicative of a failure of convergence.
 - Independent chains that produce same estimates may not necessarily have converged (pseudo-convergence)
- If we determine that the chain looks like it is in a stationary equilibrium, then this is generally accepted as evidence that the MCMC chain has converged to its stationary distribution.



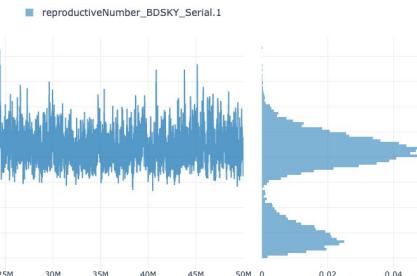
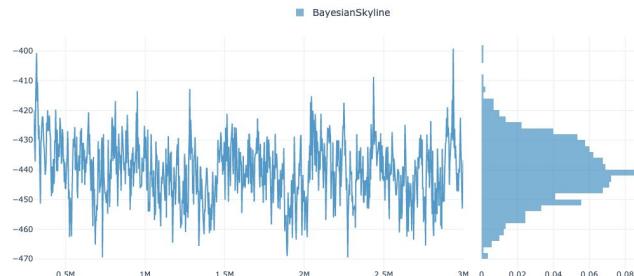
Mixing

The goals of MCMC optimization are to reduce mixing time and increase mixing quality.

- **Mixing quality** describes how well a chain is exploring a stationary distribution.
- The time taken to reach this stationary distribution is the **mixing time** of a MCMC chain.

Adjectives such as **rapid** and **good/well** are used to describe mixing time and quality respectively.

Despite our inability to confirm convergence, we can quantify and optimize the mixing quality and mixing time of the algorithm.



How long to run an analysis

“The ideal amount of time to run an MCMC analysis is the time between manuscript submission and receiving reviewers' comments (thus not delaying the paper)” - Geyer (2011)

- Assuming our MCMC is properly configured the best amount of time to run an MCMC sampler is long enough for it to converge (although this can only be guaranteed at infinity).
- A run is typically stopped once a sufficient number of samples have been collected.
- How long is long enough to achieve a sufficient sample?
 - **effective sample size (ESS)**.

Effective sample size (ESS)

- Helps to identify autocorrelation in your samples
- Auto-correlated samples are a sign of poor mixing quality and increases the mixing time.
- ESS is **not** a measure of convergence (a pseudo-converged chain can also have a high ESS).
- Can be used to determine if we have run our chain long enough, but only if we define some subjective stopping value (> 200).
- A high ESS will **not** guarantee the accuracy of our estimates (although a high ESS will increase estimate precision) and a low ESS of an uninteresting parameter does not necessarily invalidate an analysis.
- The ESS of parameters in the log file do not tell you whether the MCMC chain is mixing in phylogenetic tree space, a program like AWTY can be used to assess this (Nylander et al. 2008).
- Increase the ESS.
 - Increasing the chain length
 - Modify operators.
- The rate at which ESS increases is dependent on the mixing quality.
 - The ESS of a well mixing chain should increase linearly with the number of steps.
 - The rate of this relationship can be used to estimate how long it will take to reach a desired ESS value.

Operators

The operators are used to propose potential steps in the MCMC algorithm.

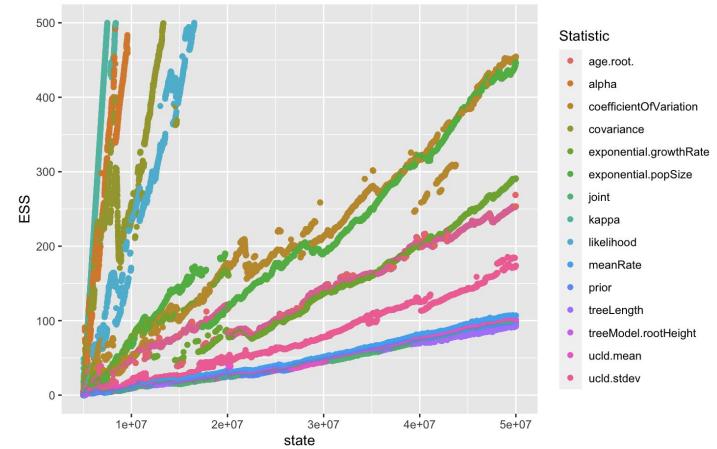
- Weight - probability of changing
- Scale - size of the change

Changing the weight of an operator increases the probability that the parameter that the operator controls will be modified in a proposed step.

ESS values generally do not grow at the same rates i.e., the ESS values of some parameters are low while others are high (cumulative ESS plots can be used to identify parameters with low growth rates).

Increase the weight on operators acting on parameters with low ESS values can improve ESS rate

BEAST will automatically optimize the values of operators by default, although extra tuning is sometimes required.



Acceptance rate

The proportion of accepted steps

Influences the number of steps required to explore the stationary distribution.

If the acceptance rate is too low or too high, the analysis will take longer to run.

- Both too high and too low acceptance rates can result in low ESS values.

In BEAST the target acceptance rate is 0.234, and by default BEAST will attempt to tune operators to reach this acceptance rate.

- Alternatively, you can adjust the jump size of operators (scale) to optimise the acceptance rate yourself.

Thinning

Thinning is the practice of throwing away every k th sample from your MCMC chain.

Thinning does not guarantee i.i.d and typically you do not need i.i.d samples

The Markov chain CLT ensures that even auto-correlated samples can be used for parameter estimation

By thinning a sample, you are just throwing away data and wasting computation.

- Conversely, thinning may lead you to falsely assume that your MCMC has good mixing quality and can reduce your effective sample size.

The only reason to thin MCMC samples is to save space.

- Typical to thin a run to a manageable size of 10,000 samples i.e. setting the sampling frequency such that 10,000 samples are collected from the analysis.

Reproducibility

Provide all resources required to reproduce the analysis, and details of checks performed to ensure the validity of the analysis.

- XML file
- The seed
- Justify your choice of priors.

Standard validity checks include

- Ensuring all parameters have ESS values of at least 200
- Confirming that multiple independent runs converge

In an ideal world all researchers using Bayesian inference would also provide the full output of their MCMC analysis as supplementary data, however, these files are often large making them difficult to distribute, and so readers generally rely on the summaries provided by the authors.