Modeling the Dynamics of Cultural Diversification

2. Introduction to LiteRate

In the last tutorial, we introduced the macroevolutionary perspective on culture as a dynamic population of cultural representations and cultural lineages that are born and die over time. We also introduced diversification rate analysis as a methodology to explain shifts in the diversity of lineages over time using evolutionary mechanisms. This included building our own diversification rate simulator that motivated the need of statistical models to capture the dynamics of underlying diversification rates.

In this module, we introduce you to the linear birth-death process, and the LiteRate algorithm built on this model. LiteRate is an unsupervised machine learning model that cuts through stochastic noise to identify statistically-significant shifts in diversification rates over time. Theoretically, these correspond with the action of evolutionary mechanisms and/or major historical events in the population's history.

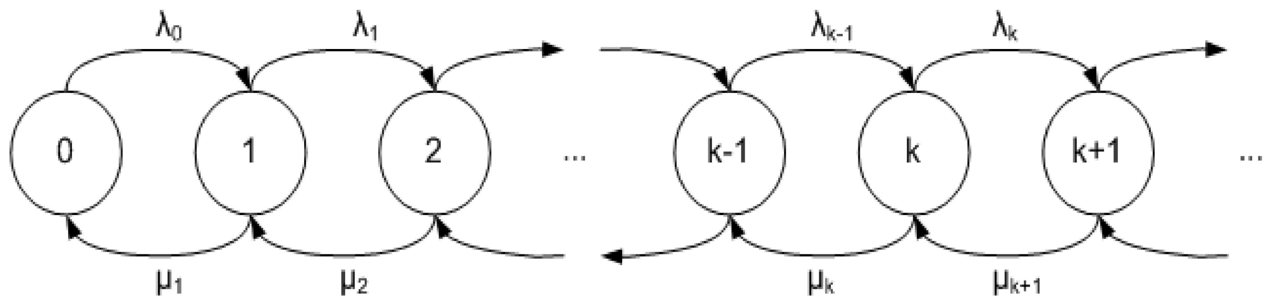
By the end of this module, you will:

- Gain an intuitive understanding of linear birth-death processes
- Be introduced to the LiteRate algorithm and its underlying MCMC sampler, Reversible Jump MCMC (RJMCMC).
- Learn how to run a LiteRate analysis.

---

a. The Birth-Death Process

The LiteRate algorithm used throughout this tutorial is built on the linear birth-death process introduced by Kendall 1948. The birth-death process is a continuous-time Markov process that defines the birth and death rates of a reproducing population over a fixed time interval. It was originally conceptualized as a particle or population moving along a line forward or backward to the next state k with instantaneous transition probabilities, which are noted as λ (per-unit birth rate) and μ (per-unit death rate).



(figure from Wikipedia)

$$\lambda_k = k\lambda \quad \mu_k = k\mu$$

Thus, two parameters describe the constant tempo/rate of origination (λ) and the tempo/rate of extinction (μ) within the population.

In the past 15 years, birth-death processes have become popular for modeling a wide variety of phenomena in Biology, and have long been used in queueing theory in Computer Science Novozhilov et al. 2006.

The likelihood of a linear-birth death process over a fixed time period given the data is defined by Keiding 1975 as:

$$P(s, e|\lambda, \mu) \propto \lambda^B \mu^D e^{-(\lambda+\mu)S}$$

where s and e are vectors of the birth and death times of individuals (e.g. lineages) in the population that lived during the time frame. B is the count of births over the time frame, D is the count of deaths, and S is the total time lived by all individuals over the time frame, summed over all individuals, lineages, or species.

$$S = \sum_{i=1}^{N} s_i - e_i$$

One intuitive way to understand this birth-death likelihood is as two generalized Poisson processes concatenated together with two events (λ^B, μ^D) and two waiting times ($e^{-\lambda S}$,$e^{-\mu S}$). For a fuller intuition on this likelihood, take the deep dive below:

▼ Deep Dive: Understanding the Linear Birth-Death Likelihood Function

A Markov process or chain is a stochastic process whose future probabilities are determined only by the previous state. Continuous-time Markov processes have the properties that the waiting time between state transitions are independent of the probabilities of the states, and that these waiting times are exponentially distributed.

Following Crawford et. al 2019, imagine that the current population size $X = k$ at time t and the waiting time until the next event (either a birth or death) is W. If the event can be either a birth or death then the waiting time $W = \tau$ is exponentially distributed with rate $\lambda_k + \mu_k$. The probability that there is a birth after waiting time $W = \tau$ is

$$P(birth, W = \tau | X(t) = k) = \left(\frac{\lambda_k}{\lambda_k + \mu_K}\right) \cdot (\lambda_k + \mu_K) e^{-(\lambda_k + \mu_k)\tau} = \lambda_k e^{-(\lambda_k + \mu_k)\tau}$$

and the probability of death is:

$$P(death, W = \tau | X(t) = k) = \mu_k e^{-(\lambda_k + \mu_k)\tau}$$

and the probability of no event is:

$$P(no\ event, W = \tau | X(t) = k) = e^{-(\lambda_k + \mu_k)\tau}$$

Now we want to determine the likelihood of the probabilities of the birth and death rates over an extended time interval (0,t), for a certain population size k during this interval. Let T_k be the total time spent in k over all visits to k, B_k be the total number of births during these visits, and D_k be the total n number of deaths. Then the total likelihood across all states k is:

$$L = \sum_{k=0}^{\infty} \lambda_k^{B_k} \mu_k^{D_k} \exp[-(\lambda_k + \mu_k) T_k]$$

We can simplify this, up to a normalizing constant, because in the linear birth death model per-unit birth/death rates are constant ($\lambda_k = k\lambda$ and $\mu_k = k\mu$):

$$L \propto \lambda^{\sum_k B_k} \mu^{\sum_k D_k} \exp[-(\lambda + \mu) \sum_{k=0}^{\infty} k T_k]$$

Now we can see the likelihood above:

- $B = \sum_k B_k =$ the total number of births over the time interval.
- $D = \sum_k D_k =$ the total number of deaths over the time interval.
- $S = \sum_k k T_k =$ the total time lived by all individuals over the time interval.

Check your understanding:

Based on how we calculated the empirical rates in the previous module for each time unit, can you intuit what the Maximum Likelihood Estimate for λ and μ are across an entire time frame?

▾ Answer:

Derived by Reynolds 1973, $\hat{\lambda}_{MLE} = \frac{B}{S} \quad \hat{\mu}_{MLE} = \frac{D}{S}$

Remember all the stochastic noise in our simulations in the previous module (Diversity and Diversification)? In our models we are going to use the above likelihood to estimate the theoretical birthrate within a time frame from the birth and death times s and e.

Programming Exercise:

In the code blocks below, calculate the linear birth-death process likelihood of the rate parameters given some observed birth and death times.

To make things a little faster, we've given you a function to calculate S called `calc_time_lived` used in the simulator in Module 2. You should try and understand it though!

We've provided two code blocks so you can write your answers in R or Python.

Because the code blocks are not editable, you should do it in a scratch cell. Create a 'Scratch code cell' ("Insert" menu-> "Scratch code cell") and copy the code from the Python or R cell below into the scratch cell. We recommend also setting the scratch code cell settings so it shows up in the bottom of the screen (3 Panels in top right corner).

Python Version (Write your answer in the notebook and press the play button to see what you get!)

```
import numpy as np

birth_times=np.array([39.01, 30.67, 41.65, 45.45, 28.67, 42.75, 40.56, 28.39, 45.39, 35.5,
death_times=np.array([50.0, 40.2, 50.0, 50.0, 48.21, 40.07, 40.96, 50.0, 50.0, 50.0, 46.16,
frame_start=10
frame_end=40
la=.12
mu=.04

def calc_time_lived(birth_times, death_times, frame_start, frame_end): #found in literate l
    '''
    calculates total time lived by all individuals within a time window
    '''
    s, e  = birth_times.astype(float), death_times.astype(float)
    s[s<frame_start] = frame_start #set elements born before timeframe to start of timefram
    e[e>frame_end] = frame_end # set elements dying after timeframe to end of timeframe.
    dt = e - s
    return np.sum(dt[dt>0])
S = calc_time_lived(birth_times, death_times, frame_start, frame_end)

###WRITE YOUR CODE HERE####
```

▼ Answer:

```
def calc_likelihood(birth_times,death_times,la,mu,frame_start,frame_end):
  B=np.sum((birth_times>frame_start) & (birth_times<frame_end))
  D=np.sum(((death_times>frame_start) & (death_times<frame_end)))
  S=calc_time_lived(birth_times,death_times,frame_start,frame_end)
  lik=la**B * mu**D * np.exp(-(mu+la)* S)
  return lik
print(calc_likelihood(birth_times,death_times,la,mu,frame_start,frame_end))
```

3.9251197773152857e-84

R Version

First, run the load_ext block first to go into R mode. Then copy the cell below that into a scratch cell

```
%load_ext rpy2.ipython
```

```
%%R

birth_times<-c(39.01, 30.67, 41.65, 45.45, 28.67, 42.75, 40.56, 28.39, 45.39, 35.5, 43.35,
death_times<-c(50.0, 40.2, 50.0, 50.0, 48.21, 40.07, 40.96, 50.0, 50.0, 50.0, 46.16, 50.0,
frame_start<-10
frame_end<-40
la<-.12
mu<-.04

calc_time_lived<-function(birth_times, death_times, frame_start, frame_end){
    # calculates total time lived by all individuals within a time window
    birth_times[birth_times<frame_start] <- frame_start #set elements born before timeframe
    death_times[death_times>frame_end] <- frame_end # set elements dying after timeframe to
    dt = birth_times - death_times
    return(sum(dt[dt>0]))
}
S <- calc_time_lived(birth_times, death_times, frame_start, frame_end)

###WRITE YOUR CODE HERE####
```

▼ Answer:

```
calc_likelihood<-function(birth_times,death_times,la,mu,frame_start,frame_end){
  B<-sum((birth_times>frame_start) & (birth_times<frame_end))
  D<-sum(((death_times>frame_start) & (death_times<frame_end)))
  S<-calc_time_lived(birth_times,death_times,frame_start,frame_end)
  lik<-la**B * mu**D * exp(-(mu+la)* S)
  return(lik)
}
print(calc_likelihood(birth_times,death_times,la,mu,frame_start,frame_end))
```

3.9251197773152857e-84

---

b. What is LiteRate?

Let's refocus on why we're doing all this! Remember our goal is to explain cultural change and stability through the rates at which cultural lineages are born and replaced over time.

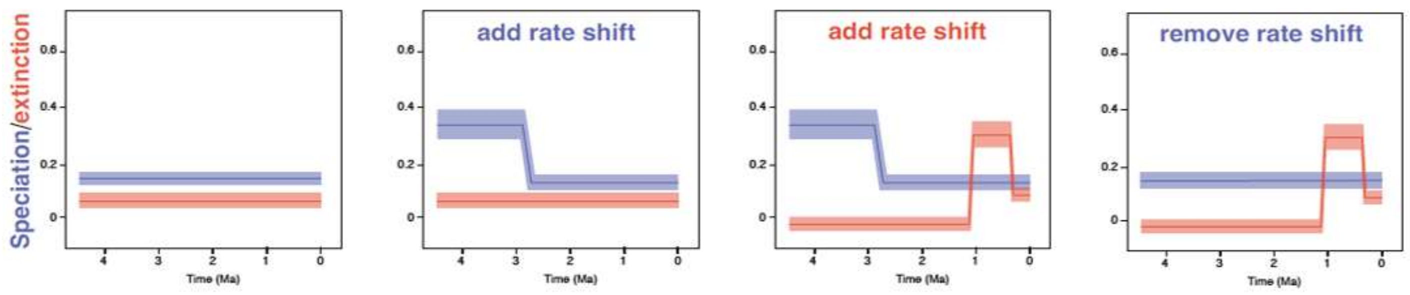
The LiteRate algorithm estimates dynamic origination and extinction rates of a population of cultural lineages from occurrence data (i.e. birth and death times). Compared to simply calculating the empirical rates in each time bin, LiteRate models diversification rates by concatenating a variable number of piece-wise constant linear-birth death processes together over time. Not only does this approach cut through noise in the empirical rates, but the statistically-significant rate shifts connecting these birth-death processes can be interpreted as consistent with evolutionary mechanisms and/or major historical events. Analysis of these statistically-significant rateshifts is a great way to assess the impact of events on the population of cultural lineages, as well as generate hypotheses about what types of evolutionary mechanisms might be operating at the macroevolutionary scale.

LiteRate is based on the PyRate algorithm (Silvestro et al. 2014; 2014; 2018), which was developed to estimate diversification rates from incomplete fossil data. By modeling data in discrete rather than continuous time and assuming that the death times of ideas/products/lineages are known, LiteRate achieves a dramatic increase in speed that allows it to scale to datasets with thousands of lineages. However if you do not have precise birth and death times or need to model in continuous time, you may want to check out PyRate. Tutorials for PyRate can be found here.

Reversible Jump Markov Chain Monte Carlo

The utility of LiteRate is that we can discover signals (i.e. statistically-significant rate shifts) in diversification rates with limited *a priori* assumptions. Under the hood, LiteRate uses a special Markov Chain Monte Carlo (MCMC) sampler called Reversible Jump MCMC (RJMCMC) to discover the number, timings, and magnitude of rate shifts from just the occurrence data (Green, 1995). From one perspective, RJMCMC can be viewed as an unsupervised machine learning approach.

Like other MCMCs, RJMCMC works by sampling proposed parameter values based on their posterior probability given the data. Posterior probabilities combine prior knowledge about the parameters (for example a rate can only take zero or positive values) with the likelihood of the data given the parameters (e.g. the probability that the lineages we observe are a product of the current values of birth and death rates). MCMCs propose new parameter values across many iterations, accepting proposals probabilistically over thousands or millions of proposals. Compared to other MCMCs, RJMCMC allows the number of parameters proposed to change over time. In LiteRate, this works by adding and removing rate shifts during proposals until the algorithm settles on a high-likelihood set of rate shifts.



For a fuller (but still intuitive) explanation of how the LiteRate likelihood expands on the standard linear birth-death process and leverages RJMCMC, see the excerpted descriptions from Koch et al., in progress in the deep dives below. For a complete description of the algorithm, see Silvestro 2019.

▼ Deep Dive: Adaptive Birth-Death Process Likelihood in LiteRate

In the LiteRate likelihood, Λ and M are now vectors of rates with length J and K corresponding to the number of time frames with different birth rates $\Lambda = \{\lambda_1, \lambda_2, \dots \lambda_J\}$, and death rates $M = \{\mu_1, \mu_2, \dots, \mu_K\}$, respectively. Similarly, B and D are now each vectors $B = \{B_1, B_2, \dots, B_J\}$ and $D = \{D_1, D_2, \dots, D_K\}$, each counting the number of birth or death events within that time frame. We also now need two new parameter vectors, $\tau^\Lambda = \{\tau_0^\Lambda, \tau_1^\Lambda, \dots, \tau_{J-1}^\Lambda\}$,and $\tau^M = \{\tau_0^M, \tau_1^M, \dots, \tau_{K-1}^M\}$ corresponding to the timings of the birth and death rate shifts, respectively. Lastly, the cumulative time lived by bands in time frame j is denoted $S_{[\tau_{j-1}^\Lambda, \tau_j^\Lambda]}$, and the cumulative time lived by bands in time frame j is denoted $S_{[\tau_{k-1}^M, \tau_k^M]}$.

$$P(\mathbf{s}, \mathbf{e} | \Lambda, M, \tau_\Lambda, \tau_M) \propto \prod_{j=1}^{J} [\lambda_j^{B_j} \times \exp(-\lambda_j S_{[\tau_{j-1}^\Lambda, \tau_j^\Lambda]})] \times \prod_{k=1}^{K} [\mu_j^{D_k} \times \exp(-\lambda_k S_{[\tau_{k-1}^M, \tau_K^M]})]$$

As an example, suppose that between 1968 and 2000 Metal had a single shift in birth rates of bands in the year 1990 and no death shifts. $J = 2, \Lambda = \{\lambda_1, \lambda_2\}$, where λ_1 is the rate from 1968-1990, $\tau^\Lambda = \{1990\}$, and $B_{j=1}$ is the total number of bands founded from 1968-1990. $S_{[\tau_0^\lambda, \tau_1^\lambda]}$ would be the total time lived by bands from 1968-1990. $K = 1, M = \{\mu_1\}$, where μ_1 is the rate from 1968-2000, and $S_{\tau_0^\mu}$ would be the total time lived by bands from 1968-2000. In LiteRate, the addition or removal of a rate shift from τ^Λ or τ^M is periodically proposed (with equal probability) throughout the chain using RJMCMC (Green, 1995).

▼ Deep Dive: RJMCMC Algorithm in LiteRate

The addition or removal of a rate shift from τ^{Λ} or τ^{M} is periodically proposed (with equal probability). When a new rate shift is added, a time window within J or K is randomly selected, and split into two. The timing of the rate shift within the chosen window is drawn from a uniform distribution, and a draw from a beta distribution is used to determine new rates that geometrically average (weighted by the length of their windows) to the old rate. Because the number of rate shifts in the model is considered unknown, we assign a Poisson distribution as a prior on J and K. The rate parameter of the Poisson prior is itself considered as unknown and assigned a Gamma hyper-prior. Lastly, the priors for the rates in Λ and M are again gamma distributions, but this time we place gamma hyperpriors on the gamma distributions' rate parameters. The use of hyper-priors makes the selection of these prior distributions less arbitrary as their shape is driven by the data.

Compared to a simple MCMC, the acceptance probability in RJMCMC is complicated by the change in dimensionality of the parameter space. Let the acceptance probability be defined as $A(\theta, \theta')$ where θ is the current set of parameters of model W, and θ' is the proposed set of parameters for model W' with an additional rate shift. In RJMCMC, the acceptance probability is thus the product of three terms: the standard *posterior ratio* and *Hastings Ratio*, but also the *Jacobian* of the parameter changes:

$$A(\theta, \theta') = \min\left(1, \quad \underbrace{\frac{\pi(\theta')}{\pi(\theta)}}_{\text{Posterior Ratio}} \quad \times \underbrace{\frac{P(W'|W)}{P(W|W')} \times \frac{P(\theta'|\theta)}{P(\theta|\theta')}}_{\text{Hastings Ratio}} \times \underbrace{\left|\frac{\partial(\theta')}{\partial(\theta, \mu)}\right|}_{\text{Jacobian}}\right)$$

The first term in the acceptance probability, the *posterior ratio*, is the ratio of the unnormalized posterior probabilities of the new state over the current state. The *Hastings ratio* consists of two terms. The first term of the Hastings ratio includes the probability of proposing a new model W' conditional on the current model W, where a model $W = \{J, K\}$ is defined by the number of birth and death rates. In our implementation we set equal probabilities to adding or removing a rate shift so that $P(W'|W) = P(W|W') = 0.5$, thus making this term equal to 1. The second term in the *Hastings ratio* is the probability of proposing a new parameter state given the current one over the opposite scenario. The final term in the acceptance probability is the *Jacobian* of the mapping function that transforms the parameters of the current state to the proposed state. This term accounts for the change in dimensionality of the parameter space. The acceptance probability of removing a rate shift is simply the inverse of the addition: $A(\theta', \theta) = A(\theta, \theta')^{-1}$.

c. Getting Started with LiteRate

What is the Data?

For LiteRate, your data should consist of a population (or representative sample) of birth and death times of lineages within a cultural form/category of interest. In some contexts, cultural lineages may be observable at multiple levels of cultural/taxonomic organization. You should pick a population where cultural lineages are distinctive, have high-consensus, and have observable birth and death times. Examples we have used in our papers include American car models that are introduced and ceased production in specific years, or Metal bands that form and break up in specific years

LiteRate Format

The LiteRate input file is a tab separated file with three columns, one row for each lineage:

- species - This column is a unique ID for the lineage (e.g., car model, band name) for your own reference.
- ts - This column refers to the time of speciation or origination and should be the earliest date at which the lineage appears.
- te - This column refers to the time of extinction and should refer to the most recent date at which the lineage was observed. **If the lineage is still extant, set this date to the last time unit in the analysis.**

Time units can be any size you like, but they must be discrete and countable. For example if your data are in months, you should recode the months as 1,2,3, etc... Time units need not start/end from 0 or 1. By default time is counted forwards (e.g 1952 occcurs later than 1951), but can also be counted backwards from the present as an option (e.g. 67 occurs later than 68).

Based on this, the typical input file for a LiteRate analysis would look something like this:

```
species ts   te
1       1952 1953
2       1952 1953
4       1958 1974
5       1958 1968
6       1968 1970
7       1978 1980
8       1961 1966
```

Requirements

LiteRate is a command line program. To use it you have be comfortable doing some minimal UNIX shell navigation. Unix commands in the notebook are preceded by a `%` or `!` Using a Unix Shell can be tricky if you've never done so before, but for this all you need to do is be able to navigate a file system. Here is a cheatsheet if you're not sure what the commands below do. If you follow commands in the notebook (roughly) on your own computer without these declamations, you should be able to download and run LiteRate on your own computer.

Using a Unix Shell

To open your UNIX shell:

- **On MacOS:** There is a built in program in MacOS called "Terminal." Open it.
- **On Windows10:** You have two options.
    - **Recommended:** Install Ubuntu on Windows. This will create a terminal/linux kernel that is coextensive with your Windows file system. **If you go this route, make sure you download Anaconda for Linux and follow the instructions to install it via command line inside your Ubuntu on Windows terminal.**
    - Alternative: Download Anaconda for Windows, and run LiteRate inside the `anacondaprompt.exe` program.
- **Linux**: You know what to do.

Installing Python + Packages

Downloading LiteRate

Now we will load LiteRate into the virtual machine that Google has provided so you can play around with the commands.

```
!git clone https://github.com/dsilvestro/LiteRate
%cd 'LiteRate'
```

If you want to install LiteRate on your own machine, run the commands above in your terminal without the `%` and `!` symbols. In order, this will:

1. Download the latest version of LiteRate into the directory from the github.
2. Change to the "LiteRate" directory.

---

d. Running LiteRate

Diversification of Metal Music Example

As an example, we are going to reproduce the main analysis from Koch et al., in progress. The dataset is the formation and breakup times of all Metal bands active between 1968-2000 in the Encyclopedia Metallum. The population is interesting in that bands can be understood as cultural lineages representing distinct musical, aesthetic, and social ideas, as well as organizations competing with each other in a field for social capital. Moreover, the dataset is remarkable because it is largely complete and manually curated: for a band to be included in the database, a recording sample must be submitted and the moderation team will determine whether it is "Metal" or "not Metal." The database has well over 120,000 bands at this point. See the paper for more details.

In order to run LiteRate, you can open a terminal window and type the following line of code.

`python3 LiteRateForward.py -d ./example_data/metal_bands/single_run/metal_bands_1.tsv -s 10000 -p 10000000 -model_BDI 2`

Note that we are not going to run LiteRate here (although you can if you want to) because 10,000,000 MCMC generations will take several hours. In this line of code, you will see that we first call on Python3 `python3`, we then call on the LiteRate program `LiteRateForward.py`. We will then specify the dataset with the flag `-d` followed by the path to the data. You will also see a number of other options that we specify, they include:

`-n` is the number of MCMC iterations. The longer you run the MCMC chain, the more likely it is to converge, but you don't want to run it longer than you have to. 10,000,000 generations is a good default.

`-s` is how frequently you sample from the chain. If you sample too frequently your samples will not be quasi-independent and your files will be enormous.

`-p` is the print frequency. It should generally be the same as `-s`.

`-model_BDI` This sets the likelihood function that LiteRate will use. The likelihood we have been working with is the Keiding Likelihood (2). The default setting (0) is an equivalent generalization that can easily be reparameterized for an immigration/death model (1). See Crawford et al., 2014 for details.

LiteRate has a number of other options as well; however, most of these options are for advanced users and will not need to be adjusted. The full list of LiteRate options can be viewed below with the following line of code.

```
!python LiteRateForward.py -h
```

---

e. LiteRate Output

After LiteRate has run to conclusion, it will create a directory called `literate_mcmc_logs` with four log files:

- `mcmc.log` This log file contains MCMC estimates for all parameters except for the rates themselves. It can be viewed in an MCMC viewer like Tracer to assess convergence.
- `sp_rates.log` This log file contains MCMC estimates for the birth rates. There are no set columns because there may be different numbers of rates in different samples.
- `ex_rates.log` This log file contains MCMC estimates for the death rates as above.
- `div.log` This log file simply counts the number of births, deaths, and standing diversity in each time unit.

Key Takeaways:

- **Birth-death processes are statistical models for estimating the underlying birth and death rates from noisy occurence data.**
- **Our statistical models are built on the likelihood of the linear birth-death process defined above.**
- **LiteRate estimates dynamic origination and extintion rates of a population of cultural lineages from occurence data (i.e. birth and death times). The algorithm identifies statistically-significant rate shifts in the data that suggest the influence of evolutionary mechanisms or historic events.**

Up Next...

In the next tutorial, we will look at how to assess convergence of LiteRate MCMC runs and interpret LiteRate results.

References

Crawford, Forrest W., Vladimir N. Minin, and Marc A. Suchard. ‘Estimation for General Birth-Death Processes’. Journal of the American Statistical Association 109, no. 506 (3 April 2014): 730–47. https://doi.org/10.1080/01621459.2013.866565.

Green, Peter J. ‘Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination’. Biometrika 82, no. 4 (1 December 1995): 711–32. https://doi.org/10.1093/biomet/82.4.711.

Keiding, Niels. ‘Maximum Likelihood Estimation in the Birth-and-Death Process’. The Annals of Statistics 3, no. 2 (March 1975): 363–72. https://doi.org/10.1214/aos/1176343062.

Kendall, David G. ‘On the Generalized “Birth-and-Death” Process’. The Annals of Mathematical Statistics 19, no. 1 (March 1948): 1–15. https://doi.org/10.1214/aoms/1177730285.

Koch, Bernard, Daniele Silvestro, and Jacob G. Foster. n.d. “The Evolutionary Dynamics of Cultural Change (as Told Through the Birth and Brutal, Blackened Death of Metal Music).” SocArXiv. osf.io/preprints/socarxiv/659bt.

Novozhilov, Artem S., Georgy P. Karev, and Eugene V. Koonin. ‘Biological Applications of the Theory of Birth-and-Death Processes’. Briefings in Bioinformatics 7, no. 1 (1 March 2006): 70–85. https://doi.org/10.1093/bib/bbk006.

Silvestro, Daniele, Nicolas Salamin, Alexandre Antonelli, and Xavier Meyer. ‘Improved Estimation of Macroevolutionary Rates from Fossil Data Using a Bayesian Framework’. Paleobiology 45, no. 4 (2019): 546–70. https://doi.org/10.1017/pab.2019.23.

Silvestro, Daniele, Nicolas Salamin, and Jan Schnitzler. ‘PyRate: A New Program to Estimate Speciation and Extinction Rates from Incomplete Fossil Data’. Methods in Ecology and Evolution 5, no. 10 (2014): 1126–1131. https://doi.org/10.1111/2041-210X.12263

Silvestro, Daniele, Jan Schnitzler, Lee Hsiang Liow, Alexandre Antonelli, and Nicolas Salamin. ‘Bayesian Estimation of Speciation and Extinction from Incomplete Fossil Occurrence Data’. Systematic Biology 63, no. 3 (2014): 349–367. https://doi.org/10.1093/sysbio/syu006