
Modeling the Dynamics of Cultural Diversification

4. Identifying Evolutionary Mechanisms in Diversification Rates

In the last tutorial we learned how to confirm, plot, and interpret LiteRate outputs. In this tutorial, we introduce you to some macroevolutionary mechanisms and their corresponding diversification rate signatures so that you can more meaningfully interpret results, use more restricted models to test hypotheses, and potentially write your own models.

In this tutorial, we will:

- Expand our simulator to allow us to model non-constant diversification processes (*section a*)
- Introduce macro-evolutionary mechanisms from Biology and simulate how they might manifest in cultural diversification rate analyses. For each of the evolutionary mechanisms listed below we provide a theoretical description, mathematical formalization, simulation, and (when applicable) statistical model (*sections b-f*):
 - Cultural stability, growth, and decline
 - Key innovations
 - Significant extinctions
 - Diversity-dependent competition
 - Environmental trends
- Introduce the DDRate program for modeling diversity-dependent competition (*section h*).
- Model competition in the Metal data (*section g*).

▼ a. Expanding our Simulator

We are now going to make a few changes to our simulator from Module 1 (Diversity and Diversification) to make it more robust. If you want to simply run these code blocks and move on to the next section you can. But if you can follow the code, you can use this simulator to develop and experiment with your own diversification rate models!

We've made the following changes from the simpler Module 1 simulator:

- We've added a simple theoretical rate generator parent class. This will be flexible to enough to accomodate a variety of theoretical models. We'll implement a simple constant rate subclass now to show you how it works.

- We are downscaling our simulations to smaller time units (e.g. simulating at the scale of days instead of years) to lessen the size of stochastic swings. In practice, this just means we multiply `time` by the scaling factor, and divide everything else by the scaling factor.
- We can rerun the simulation if it fails to meet various criteria such as:
 - A minimum number of lineages must be born
 - A maximum number of elements can be born
 - A maximum number of lineages can die
 - Anything else you want!

Everything else is the same as Module 1. If this is too overwhelming right now, just make sure you understand the rate generator class (first code block) and run the other one.

▼ Rate Generator Class

- Brief note about object-oriented programming:

```
#@title
class Rate_Generator:
    def __init__(self, param_dict, scale):
        self.param_dict=param_dict
        self.scale=scale

    def get_rates(self, population, time):
        raise NotImplementedError

class Constant_Rate_Generator(Rate_Generator): #defines it as a subclass of Rate_G
    def __init__(self, param_dict, scale):
        super().__init__(param_dict, scale) # gives us methods and properties of pare
        self.la = self.param_dict['lambda']
        self.mu = self.param_dict['mu']

    def get_rates(self, population, time):
        return self.la/self.scale, self.mu/self.scale
```

SHOW CODE

▼ b. Equilibrium

THEORETICAL DESCRIPTION: Under this hypothesis the cultural form is changing over time, but in a predictable, stable way. The theoretical standing diversity will stay flat, even if new cultural

lineages are being born and dying. This is our null hypothesis against which we can compare other models.

MATHEMATICAL FORMALIZATION: Under this hypothesis, the birth and death rates are both constant and equal. Let's formalize the birth and death rates at time t as λ_t and μ_t , for all t in frame of analysis T . Then the rates can be defined as:

$$\lambda_t = \mu_t, \forall t \in T$$

SIMULATION: The simulator below should look familiar: it is the Constant Rate Generator from

```
#@title Simulate Equilibrium
time_length=95 #@param {type:"slider", min:5, max:100, step:5}
starting_diversity= 500 #@param {type:"slider", min:5, max:1000, step:100}
theoretical_birthrate=.1 #@param {type:"number"}
theoretical_deathrate=.1 #@param {type:"number"}
random_seed=150 #@param {type:"number"}

gen_params = {'lambda':theoretical_birthrate, 'mu':theoretical_deathrate}
const_generator=Constant_Rate_Generator(gen_params, SCALE)

mySimulator=Simulator(const_generator,time_length,starting_diversity,SCALE)

results=mySimulator.run_simulation(random_seed)
plot_results(results)
```

STATISTICAL MODEL: This is the null hypothesis against which all other theoretical mechanisms can be compared (including significant shifts in `LiteRate`). You can test this hypothesis against real data as a setting in either `LiteRate` or `DDRate`.

▼ c. Key Innovations

THEORETICAL DESCRIPTION: We recognize an innovation as the birth of a cultural lineage. From a macroevolutionary perspective, we can imagine **key innovations** as those that open up a cultural form to whole new populations of people that can circulate, sustain, and reproduce it.

Possible key innovations include:

- The invention of new technologies that do not make older technologies obsolete, but simply create space for new ones. For example, the invention of regenerative-breaking created space for new hybrid car models, but did not make fully gas-powered cars obsolete.
- The innovation of completely new cultural forms. In music genres, we can imagine a rapid churn of ideas being born and dying until the sonic, aesthetic, and social parameters of the genre become clear. Once the parameters of the genre solidify, the form is named and an

audience emerges. We would then expect new artists to enter the genre following early innovators.

A Note about Cultural Carrying Capacity

Under an evolutionary lens, we can also view key innovations as expanding the **cultural carrying capacity** of a cultural form. The substance of a carrying capacity varies from question to question, but we can imagine it as the total resources available to actors to sustain the cultural form. These resources might have economic, social, and/or cognitive dimensions. For example, if our form is a population of firms in the same market, we might imagine them competing for economic resources. If our form consists of artists or scientists within a social field, we can imagine the carrying capacity in terms of social capital (Bourdieu 1996, Kahn-Harris 2006). On some level, all cultural lineages are competing for the limited time, attention, motivation, and memory of actors to reproduce, circulate, and sustain the cultural form (Koch et al, in progress). *Most of the evolutionary mechanisms described below are contingent on this concept of a cultural carrying capacity.*

MATHEMATICAL FORMALIZATION: Key innovations might appear in diversification rates as sudden jumps in the birth rates that are sustained over time due to a greater cultural carrying capacity. Let's formalize the birth and death rates at time t as λ_t and μ_t , and the baseline diversification rates as λ_0 and μ_0 . The timing of the key innovation is t_k . The net increase in birth rates at t_k due to a key innovation is λ_k .

We can now specify a simple key innovation diversification rate model as :

$$\lambda_t = \begin{cases} \lambda_0, & \text{if } t < t_k \\ \lambda_0 + \lambda_k, & \text{otherwise} \end{cases}$$
$$\mu_t = \mu_0$$

```
##@title Simulate Key Innovation
class Key_Innovation_Rate_Generator(Rate_Generator):
    def __init__(self, param_dict, scale):
        super().__init__(param_dict, scale) # gives us methods and properties of parent class R
        time_of_innov=int(self.param_dict['time_len']* self.param_dict['timing_innov'])
        self.la= np.repeat(self.param_dict['lambda'],self.param_dict['time_len'])
        self.la[time_of_innov:]+=self.param_dict['mag_innov']
        self.mu = self.param_dict['mu']

    def get_rates(self, population, time):
        return self.la[time]/self.scale, self.mu/self.scale

time_length=20 #@param {type:"slider", min:5, max:50, step:5}
timing_of_innovation=0.2 #@param {type:"slider", min:0, max:1, step: 0.1}
magnitude_of_innovation=.5 #@param {type:"number"}
starting_diversity= 500 #@param {type:"slider", min:5, max:1000, step:100}
baseline_theoretical_birthrate=.1 #@param {type:"number"}
theoretical_deathrate=.1 #@param {type:"number"}
random_seed=150 #@param {type:"number"}
```

```

gen_params = {'time_len':time_length,
              'timing_innov': timing_of_innovation,
              'mag_innov':magnitude_of_innovation,
              'lambda':baseline_theoretical_birthrate,
              'mu':theoretical_deathrate}
#SCALE is defined in hidden codeblock
key_generator=Key_Innovation_Rate_Generator(gen_params, SCALE)
mySimulator=Simulator(key_generator,time_length,starting_diversity,SCALE)
results=mySimulator.run_simulation(random_seed)
plot_results(results)

```

STATISTICAL MODEL: The simulation above is just to help you see what key innovations look like. With empirical data, we look for this signature as a statistically significant shift in `LiteRate`.

▼ d. Significant Extinctions

THEORETICAL DESCRIPTION: We recognize significant extinctions as the simultaneous death of a large number of cultural lineages. We can theorize several macroevolutionary causes for significant extinctions in different contexts. Here are some hypothetical examples (not necessarily observed in data):

- Exogenous shocks from outside the cultural system that effectively "kill" a large number of lineages. For example, the diversity of car models might respond directly to both World War II and the 1970s oil crisis.
- Transformative action by an individual. For example, we can image Trump's decision to assassinate Qasem Soleimani had a significant impact on the political hashtags circulating in the Twittersphere.
- The innovation of new cultural forms/categories that make older forms obsolete. In the Metal example, Grunge exploded as a popular music form in the early 1990s following the release of Nirvana's "Nevermind." It's possible that Grunge's rise as a popular music form caused a rapid, significant extinction of Metal bands.

In the last case, it's worth thinking about how key innovations in one cultural form (e.g. Grunge) might cause extinctions in others (e.g. Metal). This is why it's important to carefully define the cultural form that bounds your scope of analysis.

CHECK YOUR UNDERSTANDING:

MATHEMATICAL FORMALIZATION: How would you expect significant extinctions to affect death rates? How about birth rates?

▼ Answer:

We theorize significant extinctions as temporary spikes in the death rates. After a significant extinction occurs, the birth rates will subsequently jump as new lineages emerge to fill the cultural carrying capacity vacated by the extinct lineages. Although they are causally distinct, this rebound resembles a key innovation signature.

With t_e as the time of the extinction, μ_e as the net increase in deathrates, and λ_e as the bounceback in birthrates, this model could be specified as:

$$\lambda_t = \begin{cases} \lambda_0, & \text{if } t \leq t_e \\ \lambda_0 + \lambda_e, & \text{otherwise} \end{cases}$$

$$\mu_t = \begin{cases} \mu_0, & \text{if } t \neq t_e \\ \mu_0 + \mu_e, & \text{otherwise} \end{cases}$$

```

#@title Simulate Significant Extinction
class Mass_Extinction_Rate_Generator(Rate_Generator):
    def __init__(self, param_dict, scale):
        super().__init__(param_dict, scale) # gives us methods and properties of parent class R
        time_of_ext=int(self.param_dict['time_len']* self.param_dict['timing_ext'])
        self.la = np.repeat(self.param_dict['base_la'],self.param_dict['time_len'])
        self.la[time_of_ext+1:]+=self.param_dict['mag_rebound']
        self.mu = np.repeat(self.param_dict['base_mu'],self.param_dict['time_len'])
        self.mu[time_of_ext]+=self.param_dict['mag_ext']

    def get_rates(self, population, time):
        return self.la[time]/self.scale, self.mu[time]/self.scale

time_length=10 #@param {type:"slider", min:5, max:50, step:5}
timing_of_extinction=0.1 #@param {type:"slider", min:0, max:1, step: 0.1}
magnitude_of_extinction=1.0 #@param {type:"number"}
magnitude_of_birtherate_rebound=.3 #@param {type:"number"}
starting_diversity= 705 #@param {type:"slider", min:5, max:1000, step:100}
baseline_theoretical_birtherate=.15 #@param {type:"number"}
baseline_theoretical_deathrate=.1 #@param {type:"number"}
random_seed=150 #@param {type:"number"}

gen_params = {'time_len':time_length,
              'timing_ext': timing_of_extinction,
              'mag_ext': magnitude_of_extinction,
              'mag_rebound': magnitude_of_birtherate_rebound,
              'base_la':baseline_theoretical_birtherate,
              'base_mu':baseline_theoretical_deathrate}

ext_generator=Mass_Extinction_Rate_Generator(gen_params, SCALE)
mySimulator=Simulator(ext_generator,time_length,starting_diversity,SCALE)
results=mySimulator.run_simulation(random_seed)
plot results(results)

```

STATISTICAL MODEL: The simulation above is just to help you see what significant extinctions look like. With empirical data, we look for this signature as a statistically significant shift in `LiteRate`.

▼ e. Diversity Dependence

THEORETICAL DESCRIPTION: Both key innovations and significant extinctions result in high birth rates as new lineages emerge to fill unoccupied niches within the cultural carrying capacity. What happens when this carrying capacity becomes filled?

Diversity-dependent diversification or diversity-dependence describes the phenomenon where diversification rates appear to be a function of the standing diversity of the population. The macroevolutionary interpretation for this signature is that cultural lineages are competing for resources within the carrying capacity. For example:

- Car models compete for both economic and cultural resources. Obviously, economic competition exists for consumer's wallets; if a car model doesn't sell well, it will be discontinued. But competition is also operating at a cultural level for the time, memory, and attention of consumers to research options when deciding to buy a car.
- Metal bands compete for both social and cultural resources. The vast majority of Metal bands are not competing for economic capital, but they are competing for social capital within their music scene (Kahn-Harris 2006, Bourdieu 1996). Moreover, their music is competing for the limited time, attention, motivation, and memory that people have to consume, understand and reproduce Metal as a cultural form.

MATHEMATICAL FORMALIZATION: How does diversity-dependent competition manifest in diversification rates? If we suspect there is competition within a form, we would expect the birth rate to decline as the carrying capacity fills up. At the same time, increased competition will cause the death rate to increase. Eventually the carrying capacity will be saturated, and birth and death rates will converge.

We can formalize this hypothesis using only three parameters: λ_{max} (the maximum birth rate), μ_{min} (the minimum death rate), and the maximum carrying capacity K . We also need the observed D_t (standing diversity at time t).

$$\lambda_t = \lambda_{max} - \lambda_{max} * \frac{D_t}{K}$$
$$\mu_t = \mu_{min} + \mu_{min} * \frac{D_t}{K}$$

We make a slight modification to ensure that birth and death rates converge on the same value:

$$\lambda_t = \lambda_{max} - (\lambda_{max} - \mu_{min}) * \frac{D_t}{K}$$

$$\mu_t = \mu_{min} + (\lambda_{max} - \mu_{min}) * \frac{D_t}{K}$$

This model is incredibly flexible simply by changing D_t or K to something else. In one diversity-dependent model, we have explored making K dynamic following a logistic growth curve. In the model below, we essentially replace $\frac{D_t}{K}$ with an environmental trend E .

SIMULATION: Given the last two equations above, we now implement a `Diversity_Dependence_Rate_Generator` class which takes a `param_dict` with three parameters `lambda_max`, `mu_min`, and carrying capacity `K`. After you have calculated the rates at time t , we floor them so they do not become negative.

```
#@title Simulate Diversity Dependence
class Diversity_Dependence_Rate_Generator(Rate_Generator):
    def __init__(self, param_dict, scale):
        super().__init__(param_dict, scale) # gives us methods and properties of parent class R
        self.la = self.param_dict['lambda_max']
        self.mu = self.param_dict['mu_min']
        self.K = self.param_dict['carrying_capacity']

    def get_rates(self, population, time):
        la = max(0, self.la - (self.la - self.mu) * len(population.currently_alive()) / self.K )
        mu = max(0, self.mu + (self.la - self.mu) * len(population.currently_alive()) / self.K )
        return la / self.scale, mu / self.scale

time_length=50 #@param {type:"slider", min:5, max:100, step:5}
starting_diversity= 1000 #@param {type:"slider", min:5, max:10000, step:100}
carrying_capacity= 10000 #@param {type:"slider", min:1000, max:100000, step:1000}
theoretical_max_birthrate=.4 #@param {type:"number"}
theoretical_min_deathrate=.1 #@param {type:"number"}
random_seed=150 #@param {type:"number"}

gen_params = {'lambda_max':theoretical_max_birthrate,
              'mu_min':theoretical_min_deathrate,
              'carrying_capacity': carrying_capacity}
dd_generator=Diversity_Dependence_Rate_Generator(gen_params, SCALE)

mySimulator=Simulator(
    dd_generator,
    time_length,
    starting_diversity,
    SCALE
)

results=mySimulator.run_simulation(random_seed)

plot_results(results)
```


STATISTICAL MODEL: You can test this hypothesis in empirical data using `DDRate.py` (demonstrated below).

▼ f. Environmental Trends

In some scenarios, we might suspect that diversification rates within a form are responding to exogeneous phenomena more than they are driven by endogenous dynamics. To model this, we can make rates a function of trends that track these phenomena. For example,

- If we believe that the diversity of American car models is purely driven by market demand, we might correlate rates with U.S. GDP.
- If we believe that the diversity of Metal bands is purely driven by the genre's success as pop music, we might make diversity a function of band's success on the Billboard charts.

Note that in general these are very strong hypotheses, and it's more likely that rates respond to trends in certain time windows, rather than over the whole lifecourse of the form.

Mathematically, we can formulate this idea simply as:

$$\begin{aligned}\lambda_t &= \lambda_0 + \alpha * E \\ \mu_t &= \mu_0 + \beta * E\end{aligned}$$

where E is some environmental trend.

α and β can be interpreted as the strength of the linear effect of the environmental trend on birth or death rates respectively. Note that in a statistical model, we might want rates to respond more flexibly to E with linear or exponential scaling parameters. You can also change the the form of the relationship between the rates and E if you have a more specific hypothesis.

CHECK YOUR UNDERSTANDING/PROGRAMMING EXERCISE:

See if you can adapt the `Diversity_Dependence_Rate_Generator` class to model environmental trends as above. Create a 'Scratch code cell' ("Insert" menu-> "Scratch code cell") and copy the code from the extinction cell above 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 to 2 Panels).

HINTS:

- Remove the bit that forces birth and death rates to be equal.
- Add alpha and beta parameters to the rate generator.
- Make an attribute of the rate generator called `self.trend` that is an array of the trend with the same length as `time_frame`. Index this trend using the time variable in the `get_rates` function instead of the population size.

- Make your trend between 0 and 1 so rates don't get unwieldy.

The answer is in the block below. To reveal it, click on the three dots in the top right corner of the block, then "Form->show code."

SIMULATION:

```
#@title Simulate Environmental Trends
class Environmental_Trend_Rate_Generator(Rate_Generator):
    def __init__(self, param_dict, scale):
        super().__init__(param_dict, scale) # gives us methods and properties of parent class R
        self.la = self.param_dict['lambda_0']
        self.mu = self.param_dict['mu_0']
        self.alpha= self.param_dict['alpha']
        self.beta= self.param_dict['beta']
        self.trend = self.param_dict['environmental_trend']

    def get_rates(self, population, time):
        la = self.la + self.alpha * self.trend[time]
        mu = self.mu + self.beta * self.trend[time]
        return la/self.scale, mu/self.scale

time_length=40 #@param {type:"slider", min:5, max:100, step:5}
starting_diversity= 1000 #@param {type:"slider", min:5, max:10000, step:100}
theoretical_baseline_birthrate=.09 #@param {type:"number"}
theoretical_baseline_deathrate=.1 #@param {type:"number"}
alpha=.09 #@param {type:"number"}
beta=.1 #@param {type:"number"}

random_seed=150 #@param {type:"number"}

#CHANGE YOUR TREND HERE.
trend=np.array([i*2.0 for i in range(time_length+1)])
trend *= 1/trend.max() #rescale between 0 and 1 so rates don't get out of hand
time_length=len(trend)
gen_params = {'lambda_0':theoretical_baseline_birthrate,
              'mu_0':theoretical_baseline_deathrate,
              'alpha':alpha,
              'beta':beta,
              'environmental_trend': trend}

et_generator=Environmental_Trend_Rate_Generator(gen_params, SCALE)
mySimulator=Simulator(et_generator,time_length,starting_diversity,SCALE)
results=mySimulator.run_simulation(random_seed)
plot_results(results)
```

STATISTICAL MODEL: You can test this hypothesis in empirical data using `TrendRate.py`.

g. From Simulations to Statistical Models

Simulations are cool, but we'd really like to see these things in real data. Just to reiterate, below are models for testing these hypotheses in the LiteRate github. If you have any ideas about implementing models for other mechanisms, please let us know.

- Significant Extinctions (observable as significant shifts in LiteRate)
- Key Innovations (observable as significant shifts in LiteRate)
- Diversity Dependence/Competition (DDRate)
- Environmental Trends (TrendRate)

Transforming rate functions into likelihoods

When we model rates as functions of some other variable, we embed time-specific or individual-specific rates within a single linear birth-death process likelihood.

Recall that the linear birth-death process likelihood is:

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

Now let's expand that across individuals and time.

$$P(\mathbf{s}, \mathbf{e} | \lambda, \mu) = \prod_{i=1}^N \lambda(s_i) \mu(e_i) \times \exp\left(-\int_{s_i}^{e_i} \lambda_t + \mu_t dt\right)$$

Now it is easy to plug in unit-specific $(\lambda(s_i), \mu(e_i))$ or time-specific (λ_t, μ_t) rates, or both.

Estimation algorithms

We implement our own Metropolis-Hastings MCMC samplers in DDrate and TrendRate. The code is pretty straightforward if you want to adapt them. It is also easy to implement models in R and Python packages like JAGS, STAN, and PyMC3.

▼ h. Modeling Diversity Dependent Competition

▼ Setting up DDrate

Now let's return to the Metal bands example. In Module 4, we claimed that the LiteRate-estimated diversification signature for birth rates was suggestive of diversity-dependent competition. Let's test that hypothesis by (pretend) running DDrate.

First, redownload and load the LiteRate [github](#) again.

Let's check out the options for DDrate.py:

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

```
!python3 DDRate.py -h
```

Many of the options should be familiar to you from the LiteRate description in Module 3.

`-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`.

The key parameters are:

`-m_birth` To estimate constant birth rates as described in Simulation A, set this to 0. To estimate birth rates under diversity dependent competition with a fixed carrying capacity as in Simulation E set this to 1. To estimate birth rates under diversity dependent competition where the carrying capacity grows according to a logistic curve, set this to 2.

`-m_death` As above, but for death rates. In most cases, you'll want to set both the birth and death rates to the same setting.

The other parameters are either advanced or self-explanatory.

Running DDRate

Now let's pretend run the diversity dependence model on `metal_bands_1.tsv`. Again we won't run it in the notebook so we don't have to wait for the results.

```
python3 DDRate.py -d ./example_data/metal_bands/single_run/metal_bands_1.tsv -m_birth 1
-m_death 1
```

When finished, `DDRate.py` produces a single log file. This log file can be opened in Tracer and you should check the ESS and trace plots, as in Module 3.

▼ Plotting DDRate Results

The easiest way to plot the ratea from `DDRate.py` log files is to use `plotDD.py`:

```
!python plotDD.py -h
```

The key arguments for `plotDD.py` are:

- d Path to individual input file
- log_dir directory to logfile directory
- combine 0 if you wish to plot every run in `log_dir` path separately, 1 if you wish to average them.
- b burnin rate

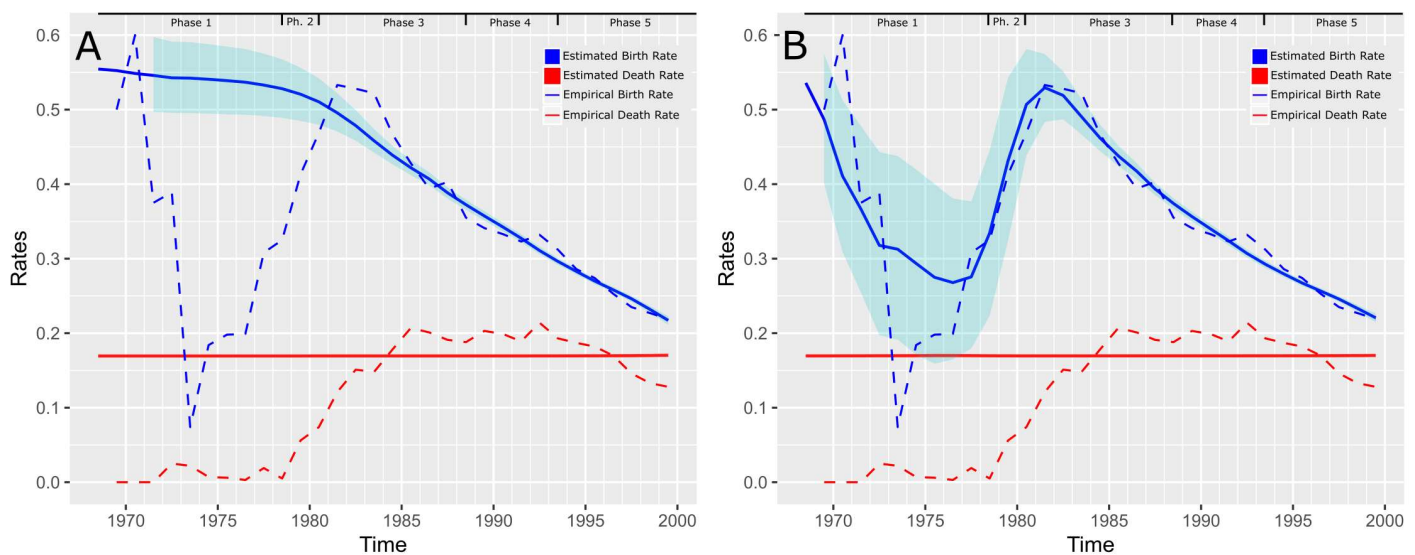
The other arguments can largely be ignored. Let's try running it on the log file we (pretend) created from running `DDRate.py`

```
!python plotDD.py -d ./example_data/metal_bands/single_run/metal_bands_1.tsv -l ./example_d
```

`plotDD.py` produces two plots: one for the empirical and estimated rates, and one for the standing diversity and carrying capacity. They are discussed and interpreted below.

g. Example: Interpreting DDRate Results using Metal Data

Below we've reproduced a figure from [Koch et al., in progress](#) produced by `plotDD.py`. Rates from the diversity-dependence models (solid lines and 95% high posterior density intervals) are plotted on top of the empirical rates (dashed lines). In Panel A, the parameter K (the carrying capacity) is an estimated parameter that doesn't change over time. In Panel B, we have made K grow parametrically according to the estimated parameters of a logistic growth curve.



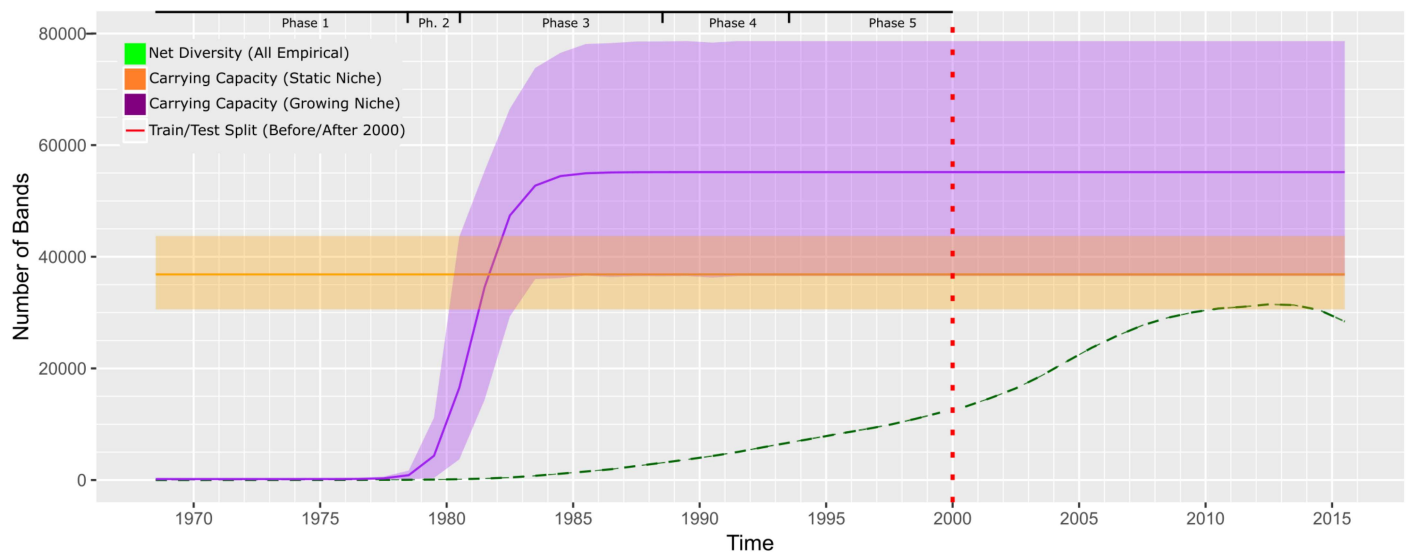
In the static-carrying capacity model, estimated birth rates seem to track empirical birth rates closely from 1987-2000. These diversity-dependent rates also track the LiteRate rates, albeit averaging over the rate shelf in Phase 4 (Figure in last module; not shown here). When we allow the carrying capacity to grow according to a logistic niche curve, the diversity-dependent rates track the observed rates roughly throughout the genre's entire history from Phase 1-5. We find this explanation compelling.

Note that neither of these models suggest that death rates are driven by competition. However death rates are comparatively much smaller than birth rates in this population, suggesting that this is a birth-driven process. This is obvious in the net diversification rates (not shown).

In this example, we interpret the carrying capacity as time, attention, motivation, and memory that actors (e.g. musicians, fans, and anyone who has some conception of the term "Metal") have to consume and develop the cultural representations and products that give the genre meaning. Collectively, these lineages bound the musical, aesthetic, and social parameters that delineate "Metal" bands from not "Metal bands." This cultural carrying capacity doesn't stymie new bands from being formed, it simply prevents them from being collectively understood by actors as Metal bands.

We model the carrying capacity as growing according to a logistic niche because it takes a critical mass of bands to concretize the parameters of the genre and spread them to an appreciable number of actors. Once this happens, the number of bands that the population of actors can collectively understand as Metal is bounded not by the number of listeners (assuredly increasing over time), but the ability of the population to share a broad understanding of what Metal means without losing coherence. See the paper for further analyses supporting this interpretation and exploring the cause of the rate shelf in Phase 4.

The cool thing about the diversity-dependence models is that they provide quantitative estimates of this theoretical carrying capacity. In the figure below, we plot these carrying capacities over time along with the standing diversity. Although the carrying capacity was estimated only from data in 1968-2000, the observed standing diversity in the dataset hasn't reached this theoretical capacity by 2016. We caution over-interpretation here because the HPDs are quite large and curatorial biases are probably depressing the standing diversity in recent years, but it is still neat to see.



Key Takeaways

- The simulator here can be flexibly subclassed to test hypotheses about evolutionary mechanisms driving diversification rates. We demonstrate simulations for significant extinctions, key innovations, diversity-dependent competition, environmental trends, and age-dependent extinction.
- Simulatable theoretical models can be easily transformed into statistical models by embedding equations into linear birth-death process likelihoods. You can then fit these likelihoods to data using maximum likelihood or simple MCMCs.
- `DDRate.py` can model diversity dependent competition. Environmental trends are modeled in `TrendRate.py`. Key innovations and significant extinctions should be observable as significant shifts in `LiteRate`.
- We show a compelling example of diversity-dependent competition under an expanding carrying capacity using the complete population of Metal bands active between 1968 to 2000.

Concluding Remarks

That's it for these tutorials! Theoretically, we hope we have convinced you of the utility of macroevolutionary perspectives that focus on cultural ideas, products, or lineages in understanding cultural change and stability over time. Empirically, we hope we have convinced you that diversification rates are useful for testing cultural macro-evolution hypotheses.

We want these tutorials to help foster a new perspective on methods useful for the macroevolutionary analysis of culture. If you have any additional questions, suggestions for these tutorials, or want to contribute models to the GitHub, please feel free to contact the lead developers: [Bernard Koch](#) or [Erik Gjesfjeld](#).

If you use these methods please cite these two papers:

Gjesfjeld, Erik, Daniele Silvestro, Jonathan Chang, Bernard Koch, Jacob G. Foster, and Michael E. Alfaro. 'A Quantitative Workflow for Modeling Diversification in Material Culture'. PLOS ONE 15, no. 2 (6 February 2020): e0227579. <https://doi.org/10.1371/journal.pone.0227579>.

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. <https://osf.io/preprints/socarxiv/659bt>.

References

Bourdieu, Pierre. *The Rules of Art: Genesis and Structure of the Literary Field*. Stanford University Press, Palo Alto 1996.

Gjesfjeld, Erik, Daniele Silvestro, Jonathan Chang, Bernard Koch, Jacob G. Foster, and Michael E. Alfaro. 'A Quantitative Workflow for Modeling Diversification in Material Culture'. PLOS ONE 15, no. 2 (6 February 2020): e0227579. <https://doi.org/10.1371/journal.pone.0227579>.

Kahn-Harris, Keith. *Extreme Metal: Music and Culture on the Edge*. Berg Publishers, Oxford 2006.

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. <https://osf.io/preprints/socarxiv/659bt>.

