

PyRate for Cultural Evolution Tutorial (Draft 1)

Erik Gjesfjeld

This is a tutorial on using the program PyRate for the analysis of cultural evolution. PyRate was originally developed by Daniele Silvestro and colleagues in a series of papers. I highly recommend reading these papers for additional details, *Methods in Ecology and Evolution* (<http://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12263/full>) and *Systematic Biology* (<http://sysbio.oxfordjournals.org/content/63/3/349.short>).

The first application of this methodology to cultural data can be found in *Palgrave Communications* (<http://www.palgrave-journals.com/articles/palcomms201619>).

Users of this tutorial should first start by creating a folder on the desktop title `PyRate_CE_Tutorial` as this folder will be where results are stored. Users should then download the PyRate program from Daniele Silvestro's GitHub site at <https://github.com/dsilvestro/PyRate> (<https://github.com/dsilvestro/PyRate>). Simply click on the clone or download button and choose the download as zip option. Unzip the folder and place the contents of the `pyrate_master` folder in the `PyRate_CE_Tutorial` folder. (You will want to see all the files, such as `PyRate.py`, `PyRateContinuous.py` and `pyrate_libraries` in the `PyRate_CE_Tutorial` folder)

This tutorial we use a combination of R (for data manipulation and plotting) and Python (run through a terminal window) for analysis. R (required) and R Studio (recommended) can be downloaded from the following websites (R (<https://www.r-project.org/>) and R Studio (<https://www.rstudio.com/home/>)). Commands that use Python will start with "python2.7" and should be run in a terminal window. All other commands will be run in R so they can be copied and pasted into the R console. (Note: This tutorial was created on and for Mac users, please adjust file paths as necessary for a Windows machine)

For more inexperienced users of the R programming environment, many functions require the installation of additional packages. For packages that are not currently installed on the user's local machine they can be installed by the following commands. (Note: a list of installed packages on the user's local machine can be found by checking for package updates under the Tools menu)

```
install.packages("name_of_package")
```

For this research the following packages will be necessary: `RCurl`; `dplyr`; `stringr`; `scales`; `laser`; `tidyr`

Once the packages have been installed, they will need to be loaded into the current R session (and reloaded every time that R is restarted). Loading libraries can be done with the following commands

```
library(dplyr)
library(tidyr)
library(stringr)
library(scales)
library(laser)
library(RCurl)
```

Getting the raw data from github

In this tutorial, we will be working with two data sets. The first contains a list of all the car models produced by American manufacturers between 1896 and 2014. This data was used in the Palgrave Communications paper (Gjesfjeld et al. 2016) (<http://www.palgrave-journals.com/articles/palcomms201619>). By going through the tutorial similar results should be obtained. The second data set contains a list of archaeological pottery ware types from the American Southwest that have had their names and dates randomized. Using the commands below, load the data into R from my github site, they data can also be downloaded manually from [www.github.com/erikgjes/PyRate_Cultural_Evolution](https://github.com/erikgjes/PyRate_Cultural_Evolution) (https://github.com/erikgjes/PyRate_Cultural_Evolution).

```
car_models <- read.csv(text=getURL("https://raw.githubusercontent.com/erikgjes/PyRate_Cultural_Evolution/master/car_models_tutorial.csv"),header=T)
pottery_data<-read.csv(text=getURL("https://raw.githubusercontent.com/erikgjes/PyRate_Cultural_Evolution/master/pottery_data_tutorial.csv"),header=T)
source("https://raw.githubusercontent.com/erikgjes/PyRate_Cultural_Evolution/master/PyRate_CE_Functions.R")
```

The first thing we want to do is look at the structure of our data. We can look at the first few entries by using the following command:

```
head(car_models)
```

```
##           make           model first_year last_year
## 1      Allstate      A-230      1952      1953
## 2      Allstate      A-240      1952      1953
## 3    AM_General      DJ5       1969      1974
## 4    AM_General      FJ8       1973      1974
## 5    AM_General    Hummer      1992      2001
## 6 American_Austin American_Austin 1930      1935
```

```
head(pottery_data)
```

```
##           Ware most_recent_age..BP. oldest_age..BP. trait
## 1 Brown_Ware_A           66           265      1
## 2 Brown_Ware_A           66           265      5
## 3 Brown_Ware_A           66           265      3
## 4 Brown_Ware_A          1130          1265      5
## 5 Brown_Ware_A          1130          1265      1
## 6 Brown_Ware_A          1165          1365      2
```

We can quickly see that the format of the two data sets is different. The car models has a date for the first and last production year of each model, or lineage. In contrast, the pottery data set has multiple dates for each type of ware. This is because these data sets are at different scales. The pottery data contains information for every pottery sherd in our analysis, whereas the car model only shows information for each car model (not every single car).

The important difference is that with the pottery data set, we have enough additional information in order to evaluate each pottery ware based on multiple dates. This is particularly useful for archaeological data where we might not be sure that the most recent and oldest ages are completely accurate. Given the detailed historical record of car models, we have high confidence that the first and last years of production for each car model is correct, so we are more interested in knowing the origination and extinction rate rather than the preservation rate. Throughout this tutorial I will refer to the pottery data set as an occurrence data set (as we have dates for each

individual occurrence) and the car models as a lineage data set (where we have dates for each lineage). Of course, we can quickly change the pottery data set from occurrence data to lineage data by summarizing the most recent and oldest dates for each ware type into a single entry (i.e. Brown_Ware_A 66 1465). However, we this should only be done if the research is highly confident in the dates being used.

Please also not that the dates in both data sets are in years before present (or BP). This means that 0 is the most recent time unit and the higher the number the further back in time. Please make sure that the your own data follows this date structure (i.e. please do not use Gregorian calendar dates).

Formatting Data For PyRate

Lineage data (car model) for PyRate

In order to use lineage data with PyRate, we must further transform our data into a structure that can be read by the program. We will use the function `lineage_pyrate`. The main output of this function is an R object and text file titled `species_pyrate_data`.

```
lineage_pyrate(car_models)
head(species_pyrate_data)
```

```
##   clade species ts te
## 1     1     307 63 62
## 2     1     308 63 62
## 3     2     861 46 41
## 4     2    1117 42 41
## 5     2    1269 23 14
## 6     3     394 85 80
```

We can see that the names of the car models and manufacturers have been replaced by numeric values. The `lineage_pyrate` function also produces two csv files which provide the keys for linking the numeric values to their names, these can be found in your desktop folder. In all of these files we are loosely borrowing from biology nomenclature by referring to the car models as species and the manufacturers of cars as clades. These terms are used in the most generic way possible with species representing our smallest taxonomic unit (car models or lineages) and clades representing a way to group species, in this case by their manufacturer.

It is possible that you may not have a adequate grouping variable (clade) for lineages. If so, simply adjust the numeric clade value to 0 using the following command.

```
species_pyrate_data$clade<-0
```

We will also want to write a text file to be used with the PyRate analysis later on.

```
write.table(species_pyrate_data, "~/Desktop/PyRate_CE_Tutorial/lineage_pyrate_data.txt", q
uote=FALSE, sep="\t", row.names = FALSE)
```

Formatting Ocurrence Data for PyRate

As discussed above, data sets in occurrence format (`pottery_data`) provide the opportunity to incorporate uncertainty about the preservation of artifacts when analyzing origination and extinction rates. If a researcher does not have a high level of certainty that the dates of first and last occurrence are correct, such as is often the case with fossils or artifacts, preservation can be informative. Please see the discussion by Silvestro et al. (2014) for further details. This function will produce an R object titled `occurrence_pyrate_data` that structures the pottery data into four categories.

```
occurrence_pyrate(pottery_data)
head(occurrence_pyrate_data)
```

```
##      Species  Status min_age max_age trait
## 1 Brown_Ware_A extinct      66     265     1
## 2 Brown_Ware_A extinct      66     265     5
## 3 Brown_Ware_A extinct      66     265     3
## 4 Brown_Ware_A extinct    1130    1265     5
## 5 Brown_Ware_A extinct    1130    1265     1
## 6 Brown_Ware_A extinct    1165    1365     2
```

The four columns in this object are: `Species`: which once again refers the smallest taxonomic unit which is each pottery sherd; `Status`: which identifies whether the sherd or ware is still used today, with archaeological data these are all extinct but could be extant with modern technological data `min_age`: this refers to the most recent age assigned to each sherd (dates should be in BP) `max_age`: this refers to the oldest age assigned to each sherd (dates should be in BP) `trait`: this refers to some discrete trait that is known about each entry. In this example, each trait could represent an archaeological site in which the sherd was found at.

This function also writes a text file directly to the `PyRate_CE_Tutorial` folder on your desktop. This text file then needs to be modified into a python readable file for analysis in PyRate. This is accomplished by calling the `pyrate_utilities.r` file which will extract the ages and create a readable python file

```
source("~/Desktop/PyRate_CE_Tutorial/pyrate_utilities.r")
extract.ages(file="~/Desktop/PyRate_CE_Tutorial/occurrence_pyrate_data.txt")
```

Additional options, such as replicates, can be included in the `extract.ages` function. See Silvestro et al. 2015 for details.

A python file titled `occurrence_pyrate_data_PyRate.py` should have been created and available in the `PyRate_CE_Tutorial` folder.

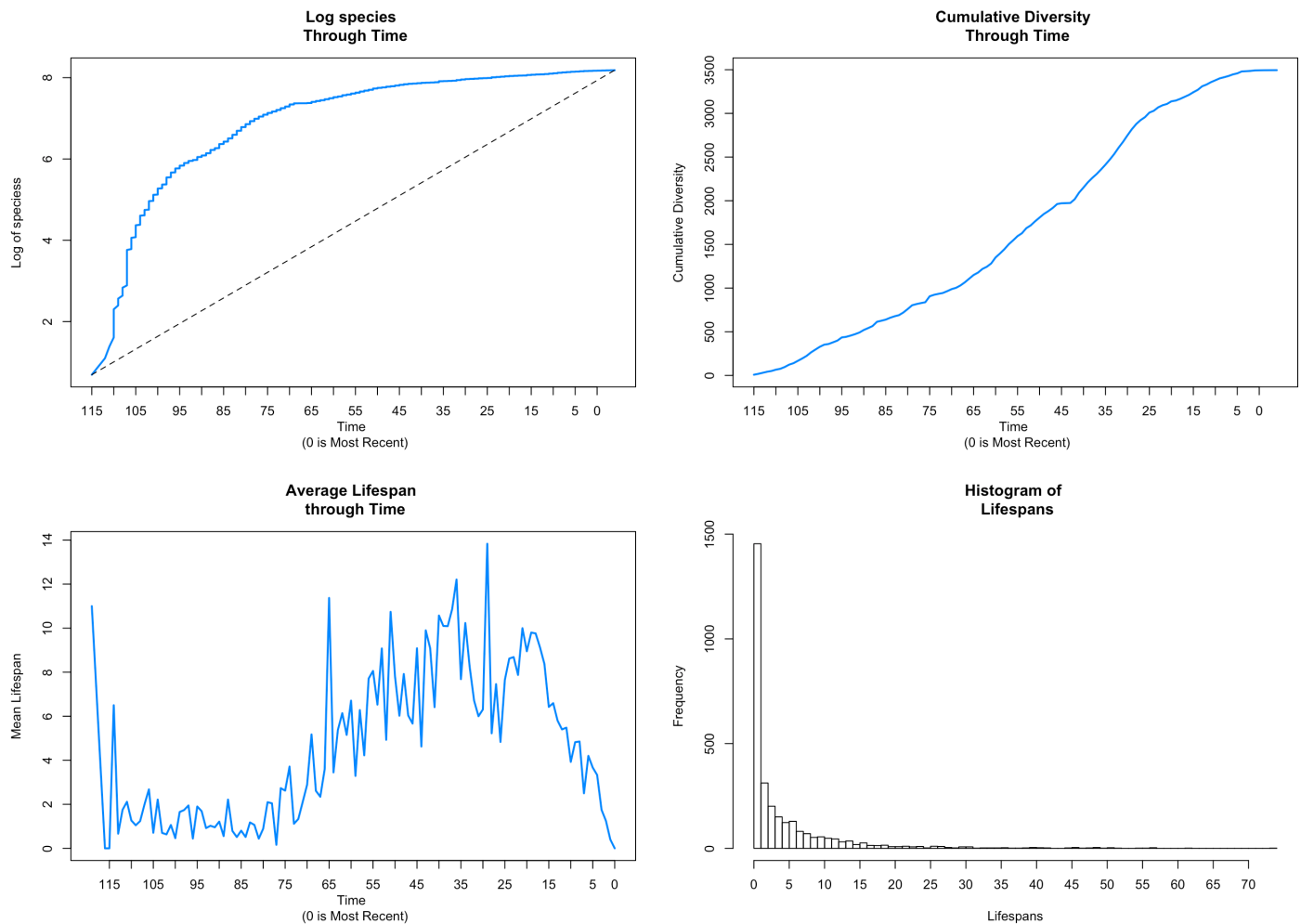
Summary Plots

Prior to starting the PyRate analysis, it is good practice to get a sense of the data by developing a series of plots. Using the function below, the following plots are created

1. Log Lineages Through Time
2. Cumulative Diversity Through Time
3. Average Lifespan Through Time
4. Histogram of Lifespans

Car Data (Lineage Data)

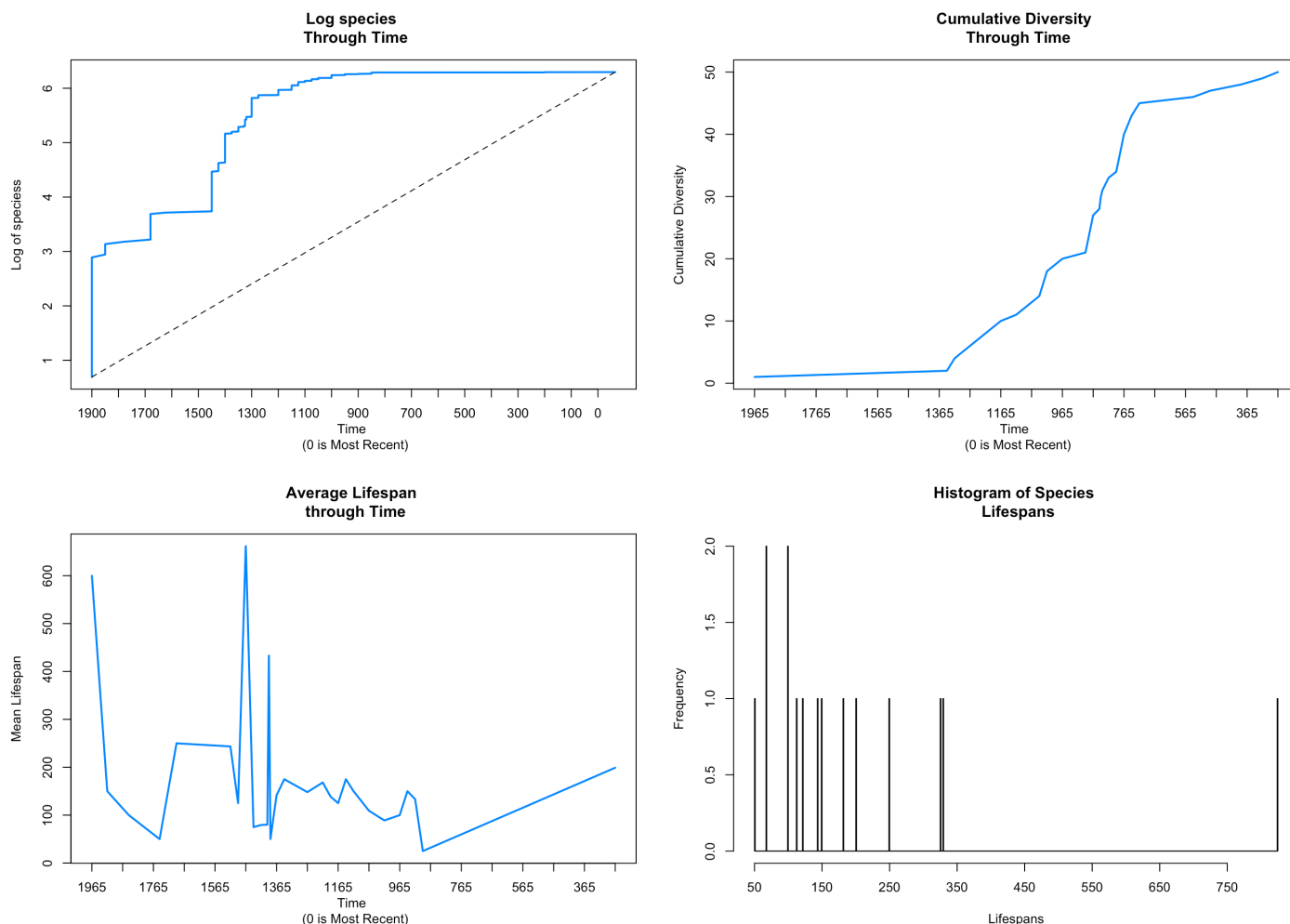
```
summary_plots(species_pyrate_data,5) #The numeral indicates the x-axis increments with the number 5 indicating 5-year time bins
```



Pottery Data (Occurrence Data)

Given the long time scales of archaeological or paleontological data, we will want to rescale the temporal bins in our graphs. Therefore, we will change the second value from 5 in the first data set to 100.

```
summary_plots_occurrence(occurrence_pyrate_data,100)
```



Choosing the best underlying model

The first decision to be made in the analysis of cultural data is whether a birth-death model is an appropriate model for the data under analysis. In this analysis, a birth-death model refers to a continuous-time Markov process where in each time step a the unit of analysis (typically a lineage or new product) can either originate (birth) or go extinct (death). Being that the birth-death model is an example of a continuous-time Markov process, the emergence or discontinuation of a lineage in the future depends on the current state of the model in the previous time step. In PyRate, if a birth-death model is favored, this means that the origination or extinction of a cultural lineage or material culture product is influenced by the existing number of lineages in the previous time step. This can be seen as suggestive of incremental innovation where the emergence of a new lineage or product is strongly influenced by the previous lineage or products available, likely through minor modifications of existing traits. A birth-death model is generally suggestive of a strong inheritance pattern between generations of cultural lineages or artifacts. (For example, a new car model that is a combination of traits from other already existing car models)

In contrast to many biological systems, where birth-death models have been shown to be very reliable, it is not always known if a birth-death model is appropriate for cultural data. An alternative model, developed for this research, is an immigration-death model. In this model, the emergence or discontinuation of a lineage or product is independent of the previous states of the model. This models has an underlying Poisson distribution which highlights that the emergence or discontinuation of a cultural lineage or product would occur randomly in each

time step. This can be seen as suggestive of more radical innovation or horizontal transmission of traits from individuals or technologies from outside of the direct lineages. (For example, a new car model that has novel traits that were developed in the airline industry).

In order to test between these two models, the data set will be analyzed in two separate analysis using the -mBDI tag. In order to execute the analysis, open a terminal window and execute the following commands.

Car Models

```
cd ~/Desktop/PyRate_CE_Tutorial
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 2 -mBDI 0 -n 10000000 #birth-death model
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 2 -mBDI 1 -n 10000000 #immigration-death model
```

Pottery Data

```
cd ~/Desktop/PyRate_CE_Tutorial
python2.7 PyRate.py occurrence_pyrate_data_PyRate.py -A 2 -fixSE null -mBDI 0 -n 1000000
0
python2.7 PyRate.py occurrence_pyrate_data_PyRate.py -A 2 -fixSE null -mBDI 1 -n 1000000
0
```

Upon executing these files, a new folder will be created in your desktop folder titled `pyrate_mcmc_logs`. This analysis uses the default settings of samples every 1,000 generations for 10,000,000 generations. Details about changes to PyRate default settings can be found in Silvestro et al. 2014. Depending on the speed of the computer the car model set will take approximately 7 hours to finish and the pottery data will take approximately 3 hours. If access to a computing cluster is available, this is the preferred option. To reduce the run time of the analysis, simply adjust the number of generations (-n) from 10 million to 1 million.

Visualizing the Rates of Origination and Extinction

Once the BDMCMC runs are finished, one can easily view the a rate through time plot by opening up a terminal window and using the following commands:

```
cd ~/Desktop/Pyrate_CE_Tutorial
python2.7 PyRate.py -plot ~/Desktop/PyRate_CE_Tutorial/pyrate_mcmc_logs/preservation_pyrate_data_1BD_marginal.rates.log

or

python2.7 PyRate.py -plot ~/Desktop/PyRate_CE_Tutorial/pyrate_mcmc_logs/occurrence_pyrate_data_1ID_marginal.rates.log
```

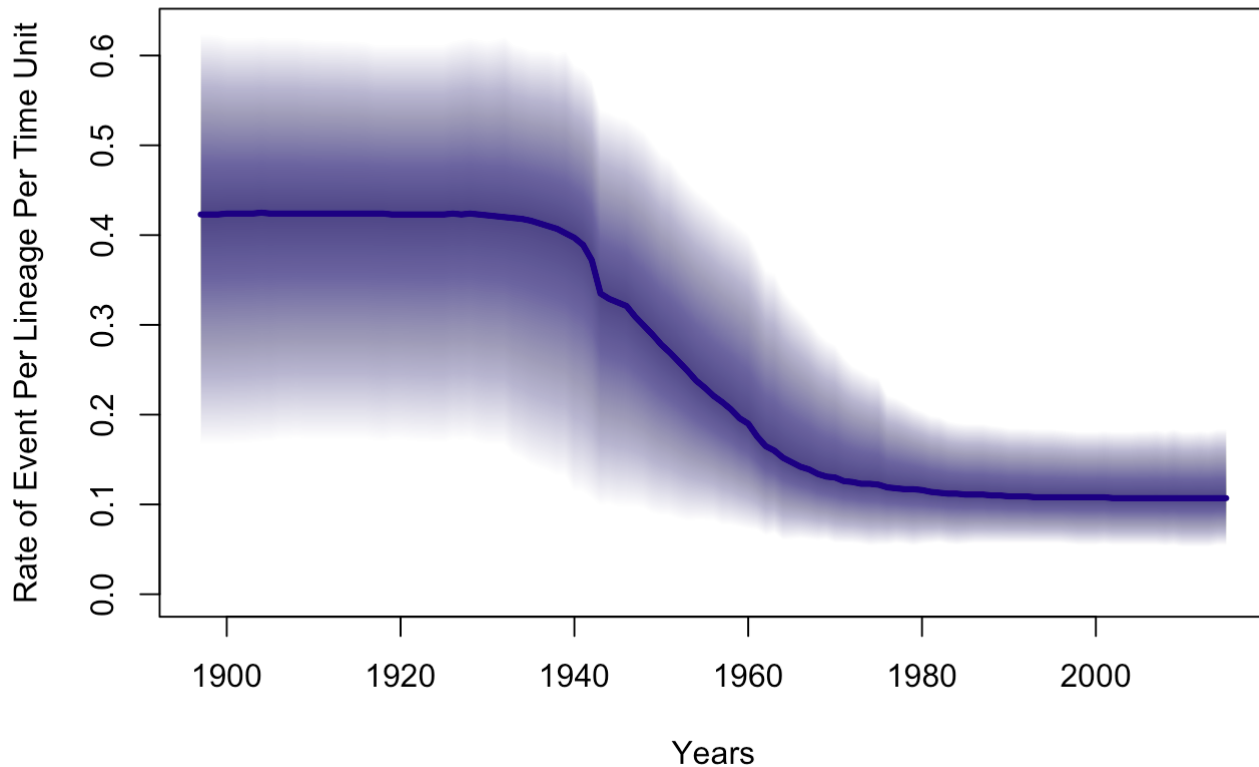
These plots will use the default plot setting in PyRate and create both an R file and pdf with the rate-through-time plots. Open the pdf and check out the rates of origination and extinction through time!

If we want to make some more refined plots we will read in the data in R and use the base packages to plot our data.

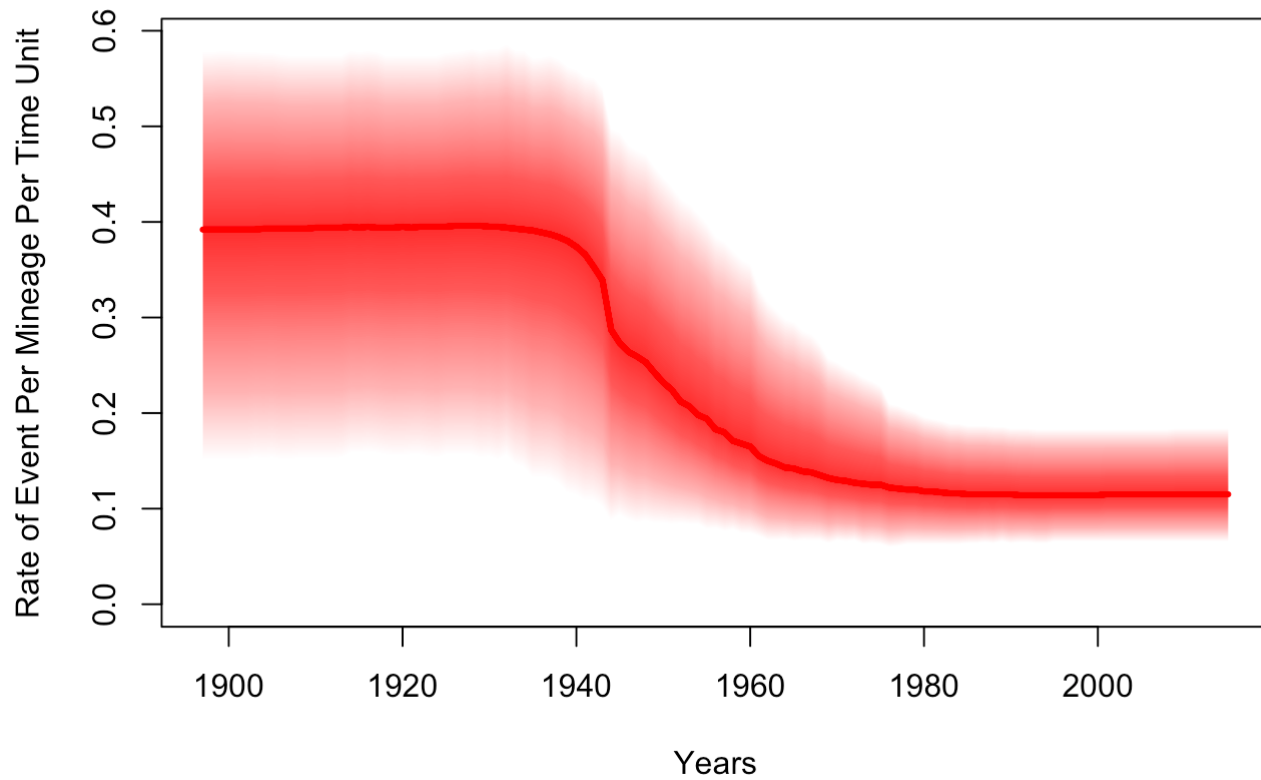
First, we will plot the lineage car model data. Note: you must run the -plot commands from the previous section before running these commands

```
source("~/Desktop/PyRate_CE_Tutorial/pyrate_mcmc_logs/lineage_pyrate_data_marginal_rates_RTT.r")
age_rev<-2015+age #adjust for the most recent date in each of the datasets

#Origination
plot(age_rev,L_hpd_M95, type = 'n', ylim = c(0,max(L_hpd_M95)),xlim = c(min(age_rev),max(age_rev)),
      xlab="Years", ylab="Rate of Event Per Lineage Per Time Unit")
plot_RTT(age_rev,L_hpd_M95,L_hpd_m95,L_mean,"navyblue")
```

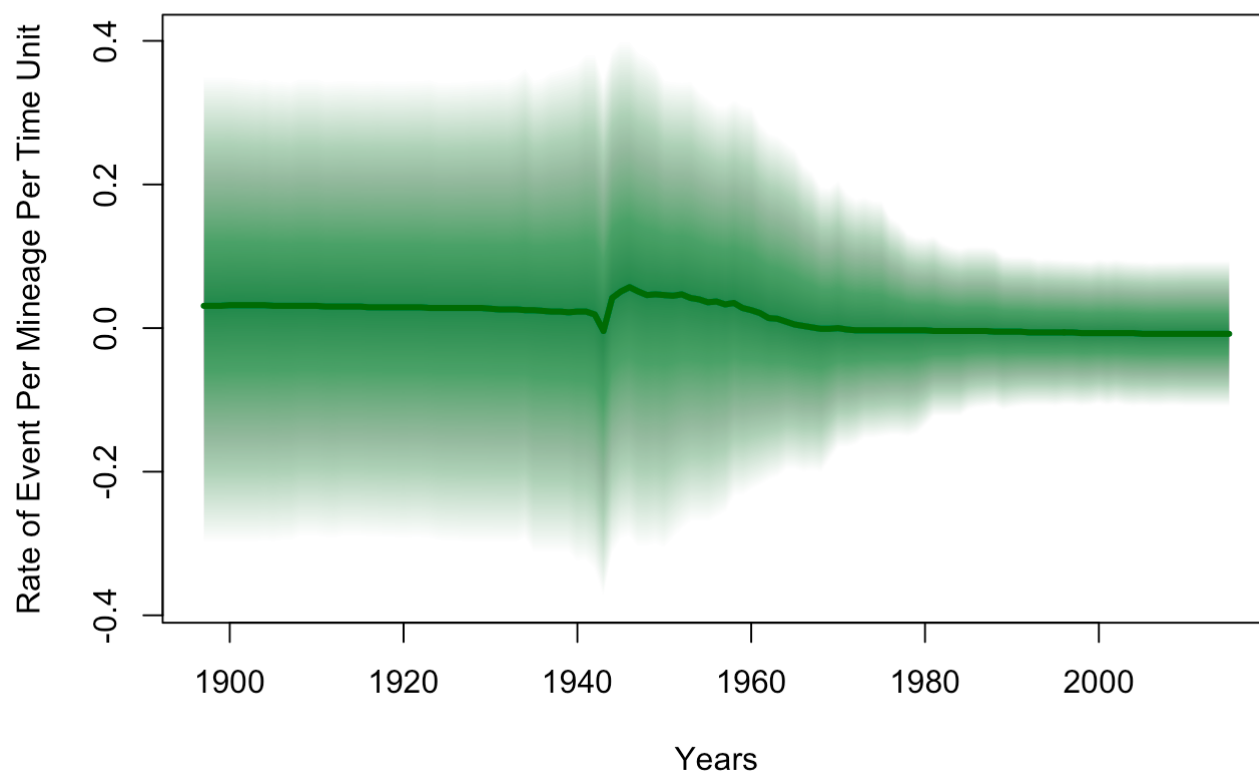


```
#Extinction
plot(age_rev,M_hpd_M95, type = 'n', ylim = c(0,max(M_hpd_M95)),xlim = c(min(age_rev),max(age_rev)),
      xlab="Years", ylab="Rate of Event Per Mineage Per Time Unit")
plot_RTT(age_rev,M_hpd_M95,M_hpd_m95,M_mean,"red")
```

```
#Net Diversification
plot(age_rev,R_hpd_M95, type = 'n', ylim = c(min(R_hpd_m95),max(R_hpd_M95)),xlim = c(min(a
e_rev),max(age_rev)), xlab="Years", ylab="Rate of Event Per Mineage Per Time
Unit",main="Net Diversification")
plot_RTT(age_rev,R_hpd_M95,R_hpd_m95,R_mean,"darkgreen")
```

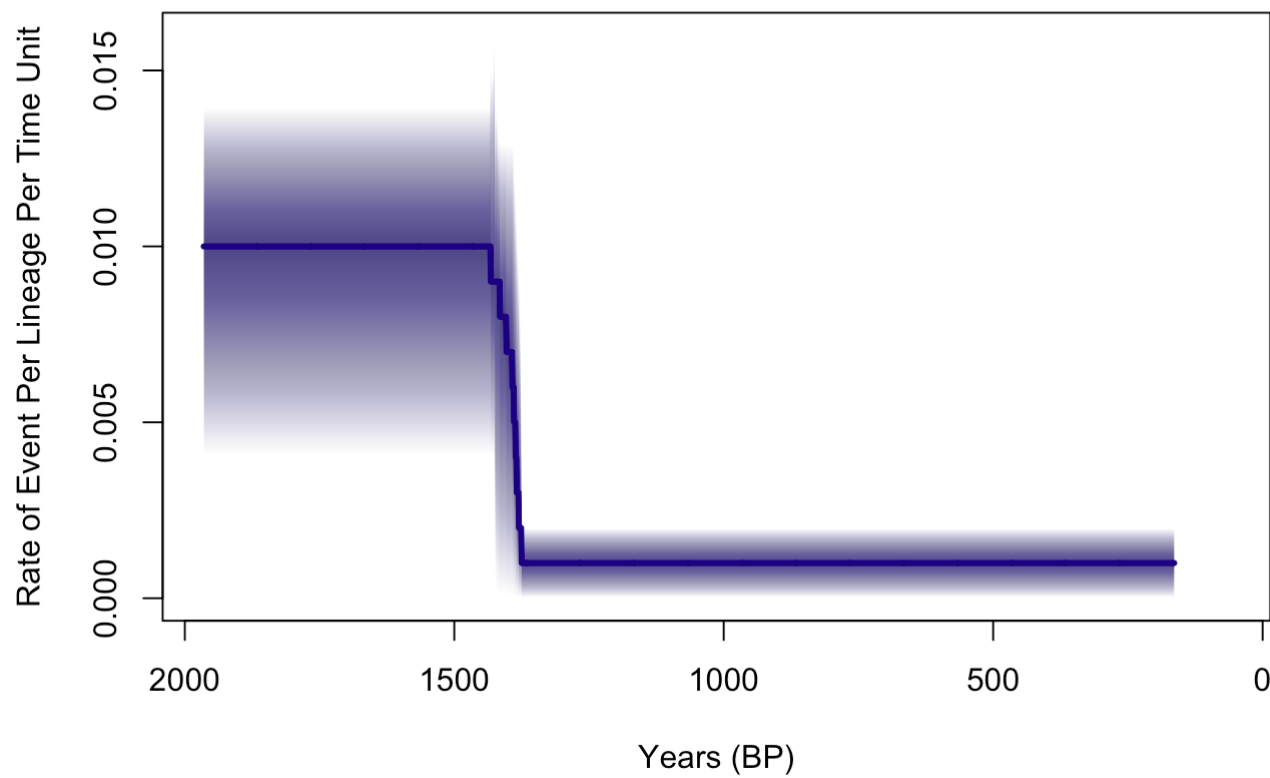
Net Diversification



And the occurrence pottery data

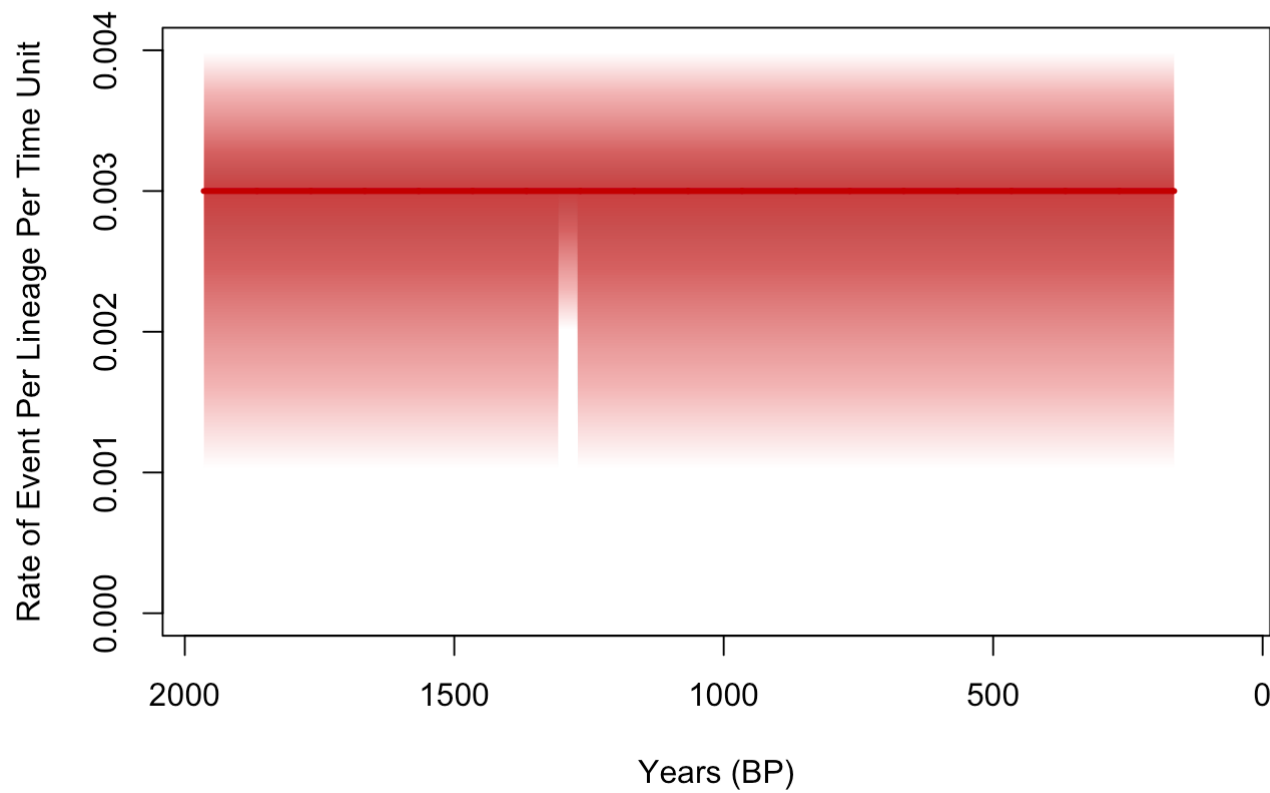
```
source("~/Desktop/PyRate_CE_Tutorial/pyrate_mcmc_logs/occurrence_pyrate_data_1BD_marginal_rates_RTT.r")
age_rev<-1965+age #adjust for the most recent date in each of the datasets

#Origination (Note that the the ages are slightly reversed in plotting as this data is plotted in BP format unlike the cars which is in a calendar format)
plot(age_rev,L_hpd_M95, type = 'n', ylim = c(0,max(na.omit(L_hpd_M95))),xlim = c(max(age_rev),min(age_rev)),xlab="Years (BP)", ylab="Rate of Event Per Lineage Per Time Unit")
plot_RTT(rev(age_rev),L_hpd_M95,L_hpd_m95,L_mean,"navyblue")
```



```
#Extinction
```

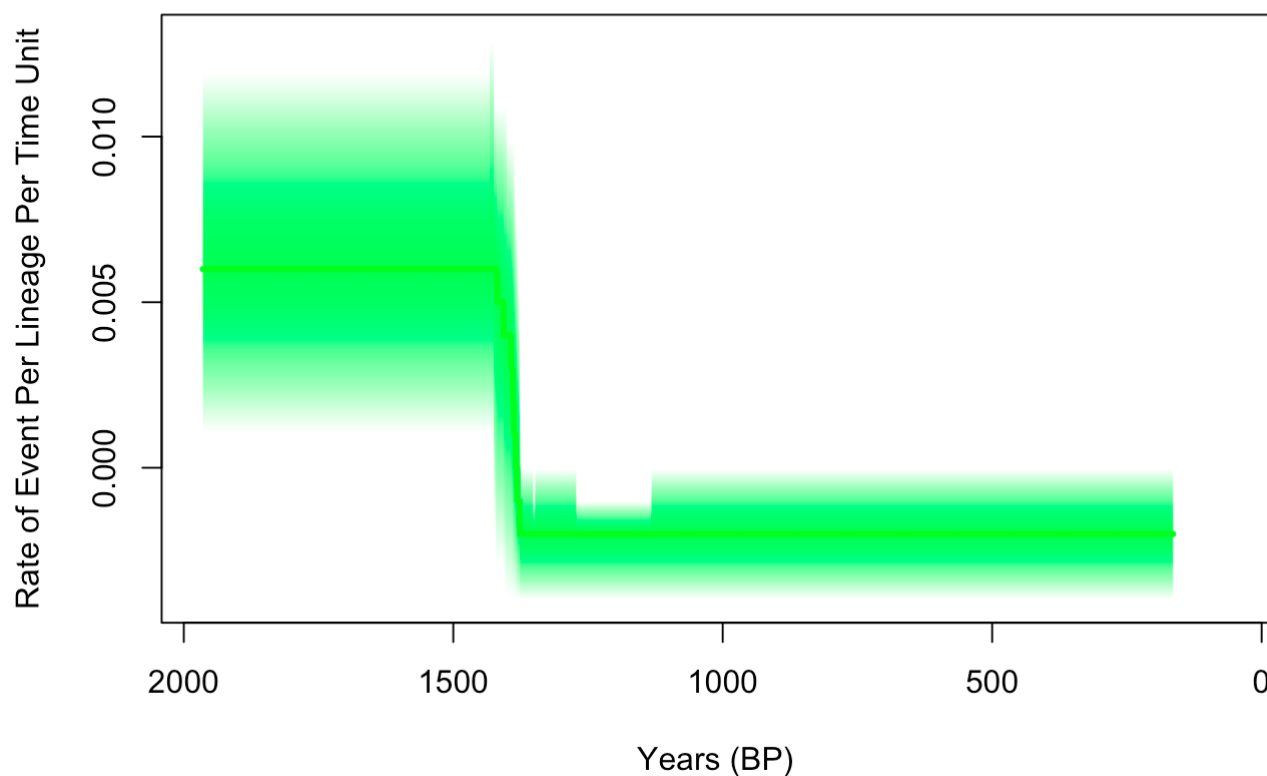
```
plot(age_rev,M_hpd_M95, type = 'n', ylim = c(0,max(na.omit(M_hpd_M95))),xlim = c(max(age_rev),min(age_rev)),xlab="Years (BP)", ylab="Rate of Event Per Lineage Per Time Unit")
plot_RTT(rev(age_rev),M_hpd_M95,M_hpd_m95,M_mean,"firebrick")
```



```
#Net Diversification
```

```
plot(age_rev,R_hpd_M95, type = 'n', ylim = c(min(na.omit(R_hpd_m95)),max(na.omit(R_hpd_M95))),xlim = c(max(age_rev),min(age_rev)),xlab="Years (BP)", ylab="Rate of Event Per Lineage Per Time Unit")
```

```
plot_RTT(rev(age_rev),R_hpd_M95,R_hpd_m95,R_mean,"green")
```



Once you have the plots available in R you can use the wide suite of graphic abilities to make publication quality graphics. Such as the figure from the Palgrave Communications publication.

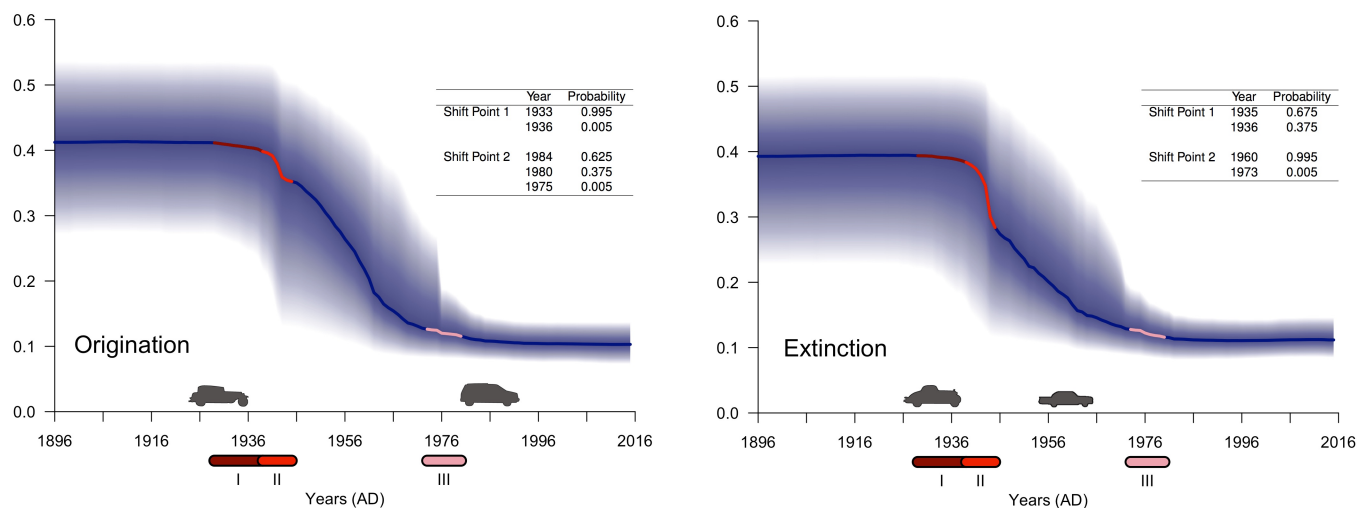


Figure 2. Origination and Extinction Rates of Car Models Through Time

It is important to note that we are unable to quantitatively test whether a birth-death or immigration-death model best fits our data from the plotting of origination, extinction and net diversification rates. In order to identify the best-fitting model we need to model test between all the most likely models.

Model testing between the BD/ID models

To determine the best fitting model, we want to first know how many different rates fit our data. It is unlikely that the rates of origination and extinction for cultural data through time are constant, but we don't know how many rates best fit the data that we have. In order to determine this, open a terminal window and enter the following commands

```
python2.7 PyRate.py -mProb <<file path to BD or ID mcmc log>>
```

Files paths are either:

```
~/Desktop/PyRate_CE_Tutorial/pyrate_mcmc_logs/occurrence_pyrate_data_1BD_mcmc.log
~/Desktop/PyRate_CE_Tutorial/pyrate_mcmc_logs/occurrence_pyrate_data_1ID_mcmc.log
~/Desktop/PyRate_CE_Tutorial/pyrate_mcmc_logs/lineage_pyrate_data_1BD_mcmc.log
~/Desktop/PyRate_CE_Tutorial/pyrate_mcmc_logs/lineage_pyrate_data_1ID_mcmc.log
```

PyRate will provide a breakdown of the probability of most likely rate models for both speciation and extinction. In addition the best configuration of rates of also provided. It is probably best to open a spreadsheet (excel or Google sheets) and record these values.

BD	Origination	Extinction	ID	Origination	Extinction
1-rate	0.0345	0.0494	1-rate	0.4375	0.0423
2-rate	0.5101	0.5104	2-rate	0.4137	0.5136
3-rate	0.3192	0.3088	3-rate	0.124	0.3074
4-rate	0.1086	0.1037	4-rate	0.0208	0.1041
5-rate	0.022	0.0227	5-rate	0.0038	0.0265
6-rate	0.0048	0.0037	6-rate	0.0002	0.0056
7-rate	0.0006	0.0009	7-rate	0	0.0004
8-rate	0.0001	0.0004	8-rate	0	0.0001
Best BD/ID			Best BD/ID		
Birth Rates	Death Rates	Rel. Prob.	Im. Rates	Death Rates	Rel. Prob.
2	2	0.268	1	2	0.224
3	2	0.158	2	2	0.215
2	3	0.154	1	3	0.138
3	3	0.101	2	3	0.122
4	2	0.053	3	2	0.062
2	4	0.051			

In this example, the best rate configuration for the Birth-Death model for the preservation data set is 2 origination rates (1 rate shift) and 2 extinction rates (1 rate shift), with a 3 origination 2 extinction rate configuration is equally likely. The best rate configuration for the Immigration-Death model is 1 origination rates (no rate shifts) and 1 extinction rates (1 rate shifts).

Once the marginal probabilities are recorded, we need to examine each of the birth-death and immigration-death configuration. We will again use PyRate but with a thermodynamic integration (TI) model rather than the birth-death MCMC to analyze the data. See Silvestro et al. (2014, 2015) for details on the implementation of a TI approach.

In the following code, -A 1 specifies the use of TI, -mL is the number of origination rates (L stands for lambda), -mM is the number of extinction rates (M stands for mu), -mBDI is the birth-death (mBDI 0) or immigration-death (mBDI 1). Each line of code should be run in a separate terminal window (or on a separate screen on a Linux server).

#This example is for lineage data only, see the example below for occurrence data.

```
cd ~/Desktop/PyRate_CE_Tutorial
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 2 -mM 2 -mBDI 0 #Most likely
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 3 -mM 2 -mBDI 0
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 2 -mM 3 -mBDI 0
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 3 -mM 3 -mBDI 0
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 4 -mM 2 -mBDI 0
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 2 -mM 4 -mBDI 0 #Least Likely

python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 1 -mM 2 -mBDI 1 #Most likely
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 2 -mM 2 -mBDI 1
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 1 -mM 3 -mBDI 1
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 2 -mM 3 -mBDI 1
python2.7 PyRate.py -d lineage_pyrate_data.txt -A 1 -mL 3 -mM 2 -mBDI 1 #Least Likely

python2.7 PyRate.py preservation_pyrate_data_PyRate.py -A 1 -mL 2 -mM 3 -mBDI 1 -fixSE null

#Occurrence Data - Only one example listed adjust accordingly for the number of different rate models as highlighted above. Note the inclusion of the -fixSE null command that is necessary to run the A1 model with occurrence data.

python2.7 PyRate.py occurrence_pyrate_data_PyRate.py -A 1 -mL ? -mM ? -mBDI 0 -fixSE null
python2.7 PyRate.py occurrence_pyrate_data_PyRate.py -A 1 -mL ? -mM ? -mBDI 0 -fixSE null
```

After the A1 models are finished (which could take a few hours) create a new folder in the PyRate_CE_Tutorial folder titled A1_mcmc_logs. Take each of the _mcmc.log files from the recently finished A1 analyses and place them in this folder. In order to compare these different rate models, we want to calculate the marginal likelihood of each model. To do this we use the following command in a terminal window

```
cd ~/Desktop/PyRate_CE_Tutorial
python2.7 PyRateContinuous.py -mL ~/Desktop/PyRate_CE_Tutorial/A1_mcmc_logs
```

PyRate continuous is a complementary program to PyRate that evaluates the correlation between our data and other variables of interest. This above command will produce a number of new files in the A1_mcmc_logs folder. These will include a “cold” version of each mcmc_log and a marginal_likelihoods.txt file. Open the marginal likelihood text file in a spreadsheet and sort the BD_lik column from largest to smallest. The model with the highest value (or closest to zero if the values are negative) is the rate model that best fits the data. In the example

data set given here, we can see that the BD_33 rate model is the best fitting model compared to all other models with a marginal likelihood of 14491.35. Furthermore, we can see that all of the birth-death models fit better than the immigration-death models suggesting that a birth-death model is the best fit for our data.

```
marg_likes<-read.delim("~/Desktop/PyRate_CE_Tutorial/A1_mcmc_log_files_tutorial/marginal_likelihooods.txt",header=T)
arrange(marg_likes,desc(BD_lik))
```

```
##               file_name replicate model   BD_lik
## 1 ebay_pyrate_american_mBDI_0_BD33_TI_mcmc      0 mcmc 14491.35
## 2 ebay_pyrate_american_mBDI_0_BD42_TI_mcmc      0 mcmc 14456.62
## 3 ebay_pyrate_american_mBDI_0_BD32_TI_mcmc      0 mcmc 14421.57
## 4 ebay_pyrate_american_mBDI_0_BD24_TI_mcmc      0 mcmc 14378.55
## 5 ebay_pyrate_american_mBDI_0_BD23_TI_mcmc      0 mcmc 14354.69
## 6 ebay_pyrate_american_mBDI_0_BD22_TI_mcmc      0 mcmc 14260.23
## 7 ebay_pyrate_american_mBDI_0_ID23_TI_mcmc      0 mcmc 14162.95
## 8 ebay_pyrate_american_mBDI_0_ID22_TI_mcmc      0 mcmc 14084.32
## 9 ebay_pyrate_american_mBDI_0_ID32_TI_mcmc      0 mcmc 14083.60
## 10 ebay_pyrate_american_mBDI_0_ID13_TI_mcmc      0 mcmc 14032.73
## 11 ebay_pyrate_american_mBDI_0_ID12_TI_mcmc      0 mcmc 13960.26
```

Finding the Times of Shift Pionts

Once we have determined the best fitting model, a 3 origination rate and 3 extinction rate model for our example data, we can identify the times that rate shifts were most likely to occur. To do this we need to run the following code:

```
BD_33<-read.delim("~/Desktop/PyRate_CE_Tutorial/A1_mcmc_log_files_tutorial/ebay_pyrate_american_mBDI_0_BD33_TI_mcmc.log",row.names=1) #This one was chosen as it was the best fitting model
BD_33<-BD_33[1:1000,] #removal of the rest of the TI as we are not interested in it

sp.1<-shift.points.func(BD_33$shift_sp_1, 2014.5) #The year is the most recent year in the dataset plus 0.5
```

```
##   max_age min_age counts proportion
## 1   1942   1942    996      0.996
## 2   1941   1941     3      0.003
## 3   1935   1935     1      0.001
```

```
sp.2<-shift.points.func(BD_33$shift_sp_2, 2014.5)
```

```
##   max_age min_age counts proportion
## 1   1975   1975   1000      1
```

```
ex.1<-shift.points.func(BD_33$shift_ex_1, 2014.5)
```



```
##    max_age min_age counts proportion
## 1    1943    1943    999      0.999
## 2    1934    1934     1      0.001
## 3    1942    1942     0      0.000
```

```
ex.2<-shift.points.func(BD_33$shift_ex_2, 2014.5)
```

```
##    max_age min_age counts proportion
## 1    1968    1968    821      0.821
## 2    1967    1967    147      0.147
## 3    1969    1969     29      0.029
```

```
shift_points<-data.frame(sp.1=sp.1[1,1],sp.2=sp.2[1,1],ex.1=ex.1[1,1],ex.2=ex.2[1,1])
shift_points
```

```
##    sp.1 sp.2 ex.1 ex.2
## 1 1942 1975 1943 1968
```

#NOTE: The number of shifts will also be one less than the number of rates expected. So, if the best fitting model is a model with 4 origination and extinction rates you will need to add a sp.3 and a ex.3 to the code above.

The resulting R object provides the years for which the rate shifts are most likely to take place. Prior to the final R object (shift points) you are also seeing the probabilities for each particular shift point.

Summary

Well, we have finally finished this tutorial! In it we have estimated the origination, extinction and net diversification rate for two cultural data sets, determined the birth-death model as the best fitting model and identified the temporal shifts points when rate shift were most likely to occur.

The broader contribution of the PyRate method, or more broadly a stratigraphic method, is the ability to evaluate the dynamics of diversification over continuous time with dates. We can see in the example of the car models that there has been a nearly four-fold decrease in the origination and extinction rate of American car models since the 1920s with the most significant decrease occurring during the 1940s and 1950s.

Finally, this is the first draft of this tutorial and any mistakes, errors or clarifications would be valuable for future drafts. Please do not hesitate to contact me at egjesfjeld@socgen.ucla.edu (mailto:egjesfjeld@socgen.ucla.edu) with any questions or comments.