[redacted]

Global sea-level and temperature changes: effects on the diversification of Pseudosuchia over macroevolutionary time scales

ABSTRACT

Present biodiversity loss warrants investigation of large-scale diversity dynamics drivers. In Pseudosuchia, researching the past could aid conservation of threatened members. Amongst macroevolutionary diversity-shaping environmental parameters, sea-level and temperature are highlighted. Previous work points to differing temperature and sea-level effects on terrestrial and marine speciation and extinction rates. This study aims to analyse global and macroevolutionary effects of these factors on terrestrial and marine Pseudosuchia to better characterize their diversification. Agreeing with the literature, global temperature increased terrestrial speciation and did not affect marine extinction rates; and global sea-level rise negatively and positively correlated with terrestrial extinction and speciation, respectively. No correlation between temperature and terrestrial extinctions, and sea-level and marine speciation were controversial, suggesting fossil bias, or lower temperature effects for the former. Interestingly: no correlations between marine extinction and sea-level and positive marine speciation-temperature correlation. Ectothermic physiology and habitat restrictions/expansions underly obtained results.

INTRODUCTION

One way to understand how biodiversity is presently changing is by investigating the factors that shaped it in the past. Considering anthropogenic climate change (Rosenzweig et al., 2008) and decreasing biodiversity (Bellard et al., 2012), such research is especially relevant. In the case of Pseudosuchia, an ancient reptile group that includes extant crocodilians and their extinct relatives (archosaurs closer to them than to birds) (Mannion et al., 2015), investigating drivers of its diversification could help conserve threatened members (iucncsg.org, 2021). Numerous factors are suggested to have molded the morphologically and ecologically more diverse past pseudosuchians into their present forms. Abiotic factors include changes in temperature (Shirley and Austin, 2017; De Celis, Narváez and Ortega, 2020; Dunne et al., 2021), sea level (Mannion et al., 2015; Tennant, Mannion and Upchurch, 2016), and aridity (Carvalho et al., 2010; Mannion et al., 2015). Biotic factors include competition (Benton, 2009) and "post-extinction opportunism" (Mannion et al., 2015). At this macroevolutionary scale, environmental changes likely played leading roles (Benton, 2009), and temperature and sea level are the most cited. It is agreed that higher temperatures potentially led to higher diversity in terrestrial Pseudosuchia, and lower

[redacted]

diversity when temperatures decreased (Markwick, 1998; Carvalho et al., 2010; Mannion et al., 2015; Shirley and Austin, 2017; De Celis, Narváez and Ortega, 2020; Dunne et al., 2021). Effects of temperature on marine species diversification are debated as either the same (Martin et al., 2014) or none (Mannion et al., 2015). Pertaining sea level, higher and lower global levels seemingly led to increased and decreased marine pseudosuchian biodiversity, respectively (Mannion et al., 2015; Tennant, Mannion and Upchurch, 2016). Less studied in terrestrial pseudosuchians, sea level could be inversely correlated with speciation and positively with extinction (Klausen, Paterson and Benton, 2020). The present study investigated effects of global sea level and temperature change through time on speciation and extinction rates of terrestrial and marine Pseudosuchia, to better characterize these potential environmental drivers of diversification for this clade.

RESULTS

1. Higher temperatures drive speciation of terrestrial Pseudosuchia in macroevolutionary time-scales

Hypothesis

Speciation rates in terrestrial Pseudosuchia and global temperature will be significantly positively correlated.

Statistical testing

To test this hypothesis, the correlation coefficient between a time series of global temperatures (**Figure 1**) and a speciation rate time series of terrestrial Pseudosuchia(**Figure 2**)(See **Figure 3** for phylogeny from which rates were modelled) was calculated.***

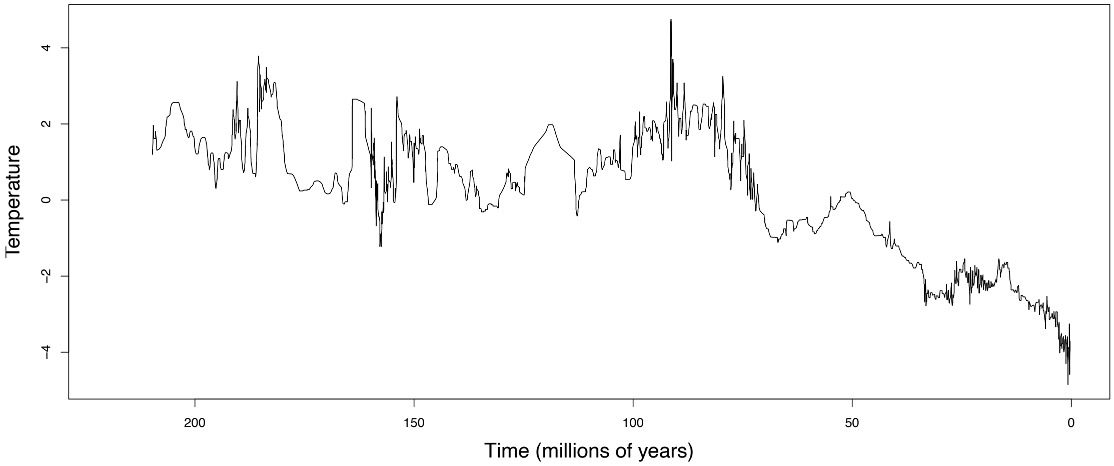


Figure 1: Global temperature through time*. Relative global temperature from paleoenvironmental proxies (marine microfossils) presented as a ratio and plotted against geological time (in millions of years) from ~210 million years ago, to present. Data from Veizer et al. (1999) and Zachos et al. (2001).*

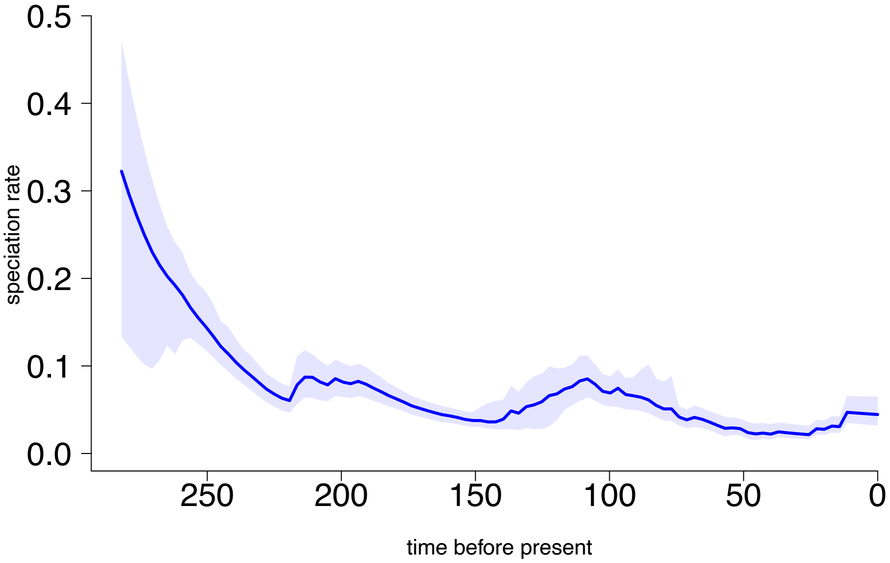


Figure 2: Speciation rate of terrestrial Pseudosuchia through time. *Terrestrial Pseudosuchia speciation rate (modelled from the group's phylogeny and presented in species per million years) plotted against geological time (in millions of years) from ~282 million years ago, to present. The solid blue line and lighter blue shaded area correspond to the speciation rate through time and confidence intervals, respectively. Data from Payne et al. (In prep).*

[redacted]

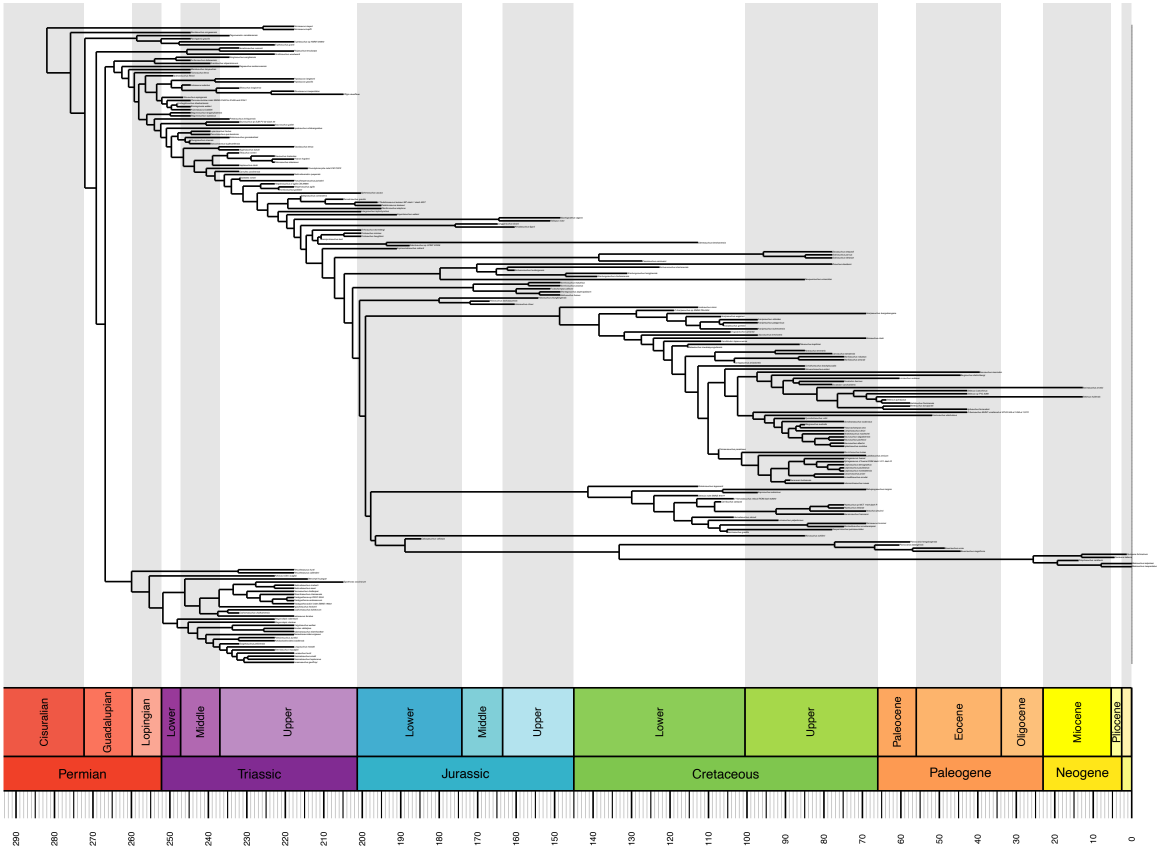


Figure 3: Phylogenetic tree of terrestrial Pseudosuchia scaled to geological time. *Phylogenetic tree of terrestrial Pseudosuchia (including 207 species and built using DNA sequence data from extant species and fossil data from extinct species) calibrated to geological time (shown in millions of years, periods and epochs) and spanning from ~282 million years ago to present. Data from Payne et al. (In prep).*

Result

There was a significant positive correlation (r=0.267±0.101) between terrestrial Pseudosuchia speciation rate and temperature (Wilcoxon unpaired test:p=3.96x10^{-18}; 95% CI[0.062, 0.428])(**Figure 4**).

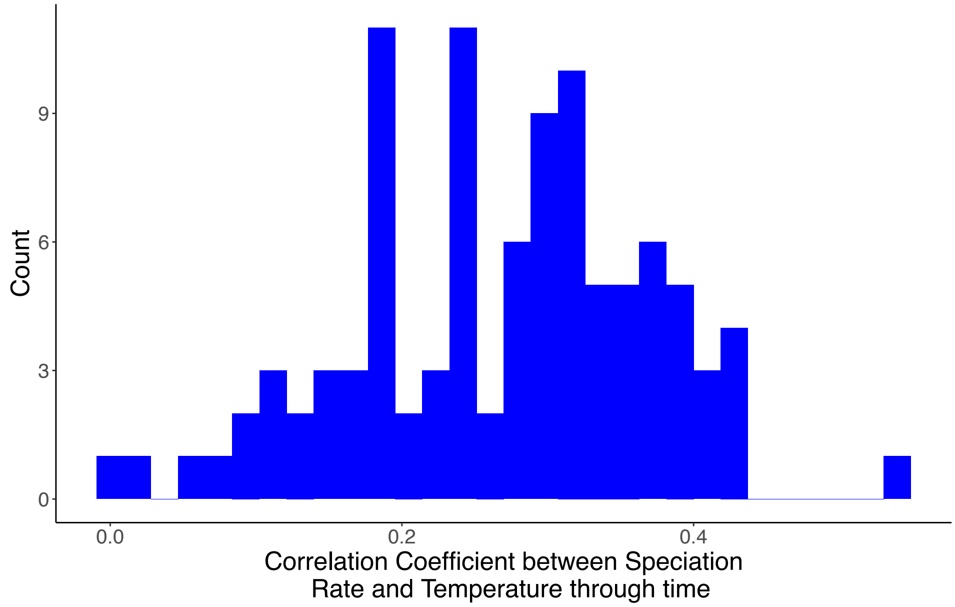


Figure 4: Correlation coefficient between terrestrial Pseudosuchia speciation rate and temperature through time. *Histogram showing the distribution of the terrestrial Pseudosuchia speciation rate and temperature through time correlation coefficients (mean = 0.267 ± 0.101).*

Interpretation

The hypothesis is accepted, in line with previous research. Previous evidence included warmer-temperature-restricted geographic distributions (Carvalho et al., 2010; Mannion et al., 2015; De Celis, Narváez and Ortega, 2020; Dunne et al., 2021), and historic biodiversity increases and range expansions during warm periods (Mannion et al., 2015; Dunne et al., 2021). This could be due to the Pseudosuchians' presumed (de Ricqlès, Padian and Horner, 2003; Cubo et al., 2020; Dunne et al., 2021) but contested (Legendre, 2014; Legendre et al., 2016) ectothermic physiology. Further research on extinction rates of terrestrial pseudosuchians is suggested.

2. Macroevolutionary time-scale extinctions of terrestrial Pseudosuchia and temperature

Hypothesis

Extinction rates in terrestrial Pseudosuchia will be negatively correlated to temperature.

Statistical testing

To test this hypothesis, the correlation coefficient between a time series of global temperatures (**Figure 1**) and an extinction rate time series of terrestrial Pseudosuchia (**Figure 5**)(modelled from phylogeny(**Figure 3**)) was calculated.***

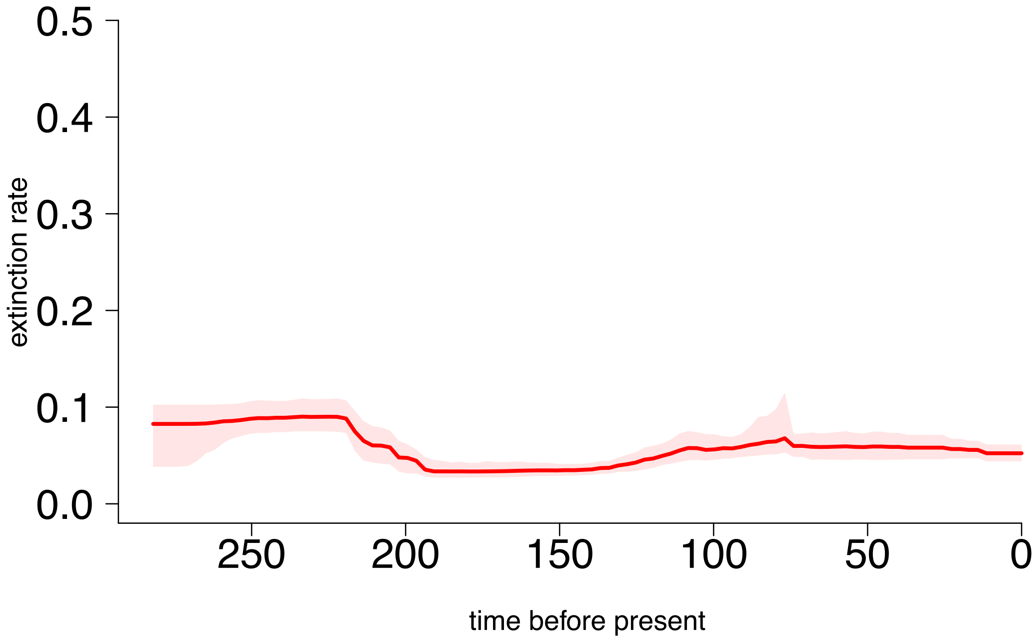
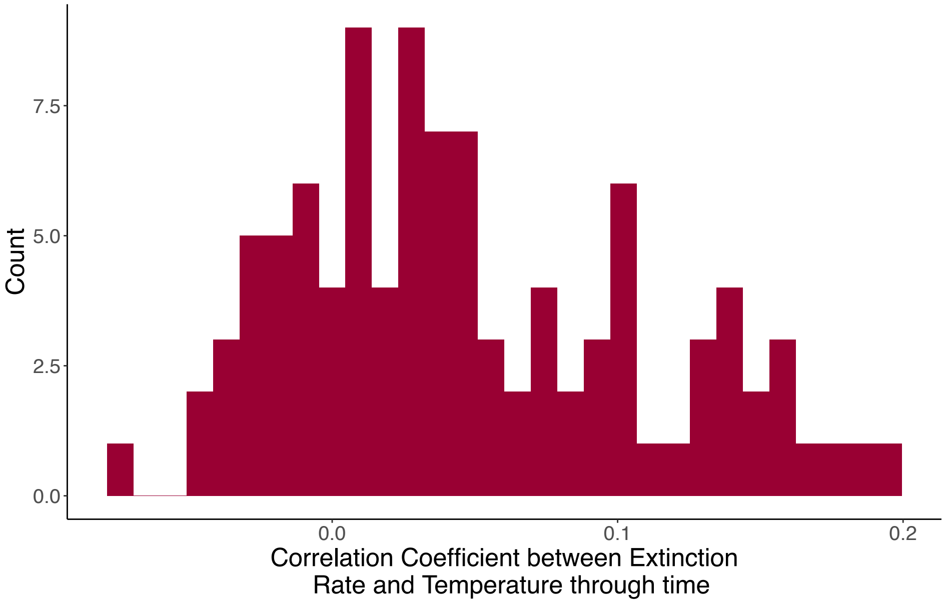


Figure 5: Extinction rate of terrestrial Pseudosuchia through time. *Terrestrial Pseudosuchia extinction rate (modelled from the group's phylogeny and presented in species per million years) plotted against geological time (in millions of years) from ~282 million years ago, to present time. The solid red line and lighter red shaded area correspond to the extinction rate through time and confidence intervals, respectively. Data from Payne et al. (In prep).*

Result

There was no correlation (r=0.0475±0.0617) between terrestrial Pseudosuchia extinction rate and temperature (Wilcoxon unpaired test: p=4.13x10-10; 95% CI[-0.0417, 0.172])(**Figure 6**).



[redacted]

Figure 6: Correlation coefficient between terrestrial Pseudosuchia extinction rate and temperature through time. *Histogram showing the distribution of the terrestrial Pseudosuchia extinction rate and temperature through time correlation coefficients (mean = 0.0475 ± 0.0617).*

Interpretation

Results reject hypothesis. This is not supported by the literature, in which all cases highlighted a negative correlation between temperature and extinctions (Markwick, 1998; Mannion et al., 2015; Shirley and Austin, 2017; De Celis, Narváez and Ortega, 2020). This could suggest a less dramatic effect of cooling temperatures. Although time-calibrated phylogenies, used here, are highly reliabile, fossil record bias is innate to macroevolutionary studies (Dunne et al., 2021). Further studies on pseudosuchian diversification could add another perspective.

3. Macroevolutionary-time-scale speciation of marine Pseudosuchia and temperature

Hypothesis

There will be no correlation between marine Pseudosuchia speciation rates and global temperature.

Statistical testing

To test this hypothesis, the correlation coefficient between a time series of global temperatures (Veizer et al., 1999; Zachos et al., 2001)(**Figure 1**) and a speciation rate time series of marine Pseudosuchia (**Figure 7**) (See **Figure 8** for the phylogeny from which rates were modelled) was calculated.***

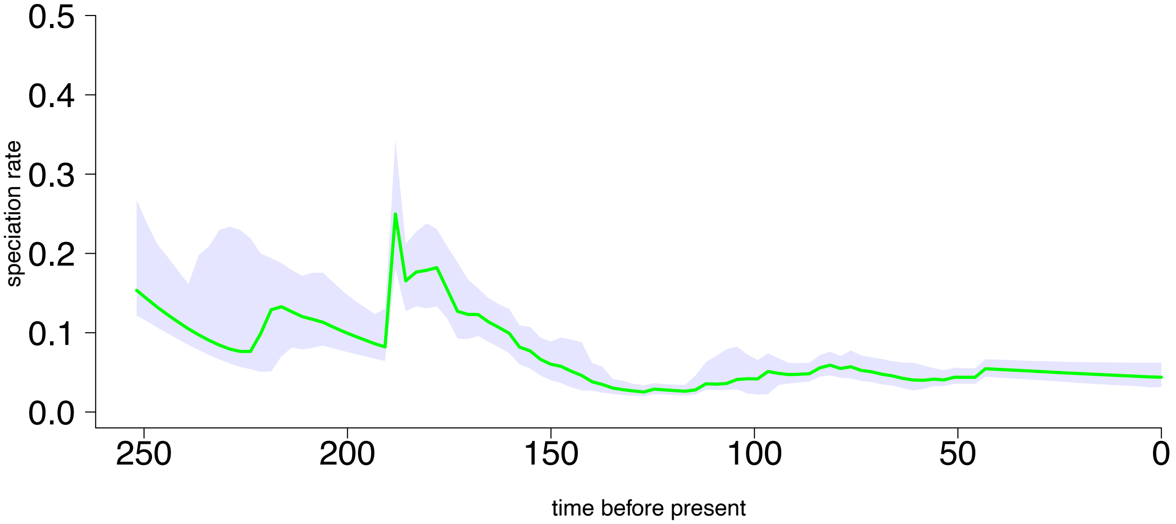


Figure 7: Speciation rate of marine Pseudosuchia through time. *Marine Pseudosuchia speciation rate (modelled from the group's phylogeny and presented in species per million years) plotted against*

[redacted]

geological time (in millions of years) from ~252 million years ago, to present time. The solid green line and light blue shaded area correspond to the speciation rate through time and confidence intervals, respectively. [*Plot is displaced by ~6.29 million years due to an issue with BAMMtools (Rabosky et al., 2014) making it impossible to adjust in R studio. Times presented in the plot should be read as ~6.29 units larger, making the time frame ~258 million years ago to ~6.29 million years ago]. Data from Payne et al. (In prep).*

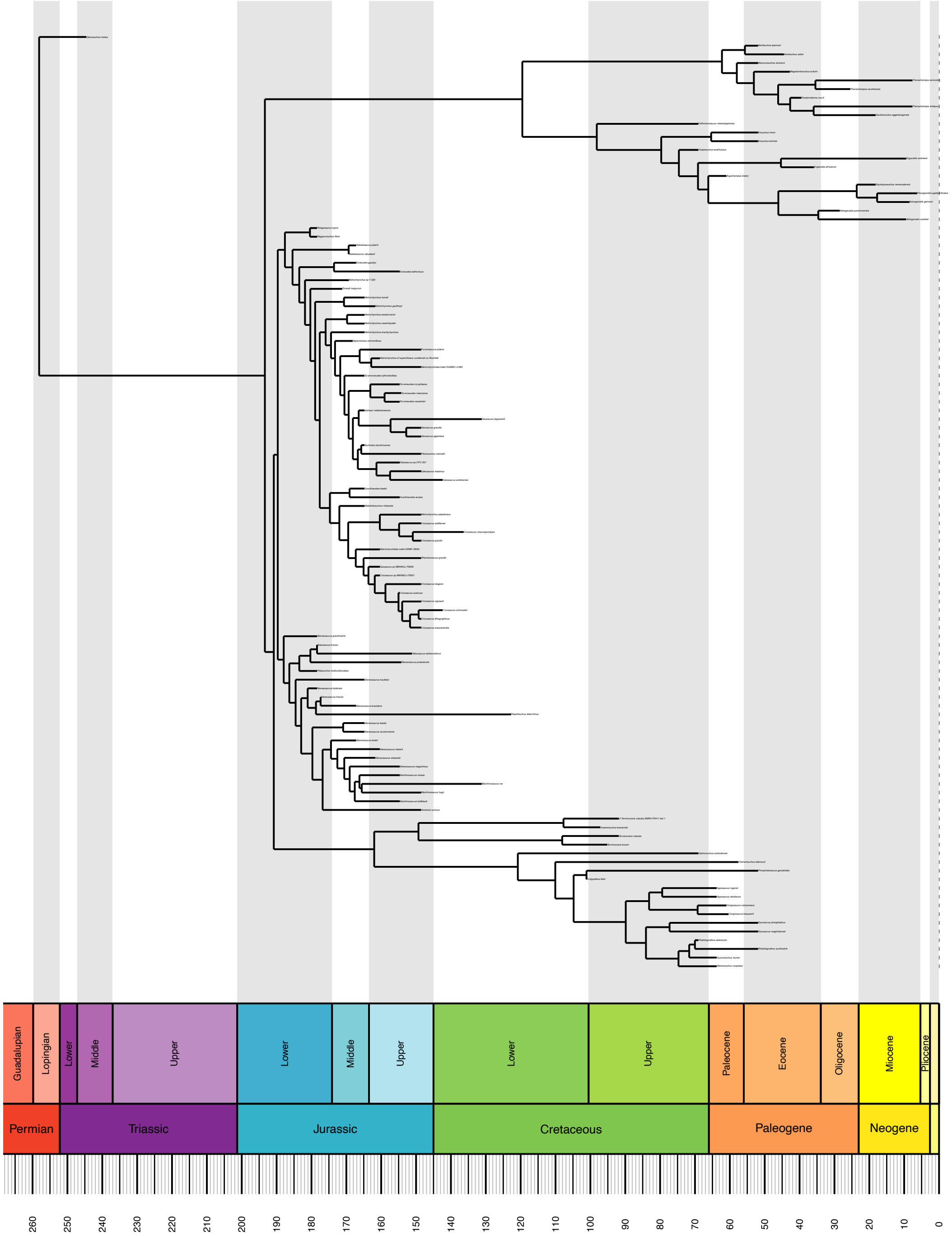


***Figure 8: Phylogenetic tree of marine Pseudosuchia scaled to geological time**. Phylogenetic tree of marine Pseudosuchia (including 108 species and built using fossil data) calibrated to geological time*

(shown in millions of years, periods and epochs) and spanning from ~258 million years ago to ~6.29 million years ago. Data from Payne et al. (In prep).

Result

There was a significant positive correlation (r=0.472±0.0614) between marine Pseudosuchia speciation rates and temperature (Wilcoxon unpaired test: p=3.95x10-18; 95% CI[0.348, 0.584])(**Figure 9**).

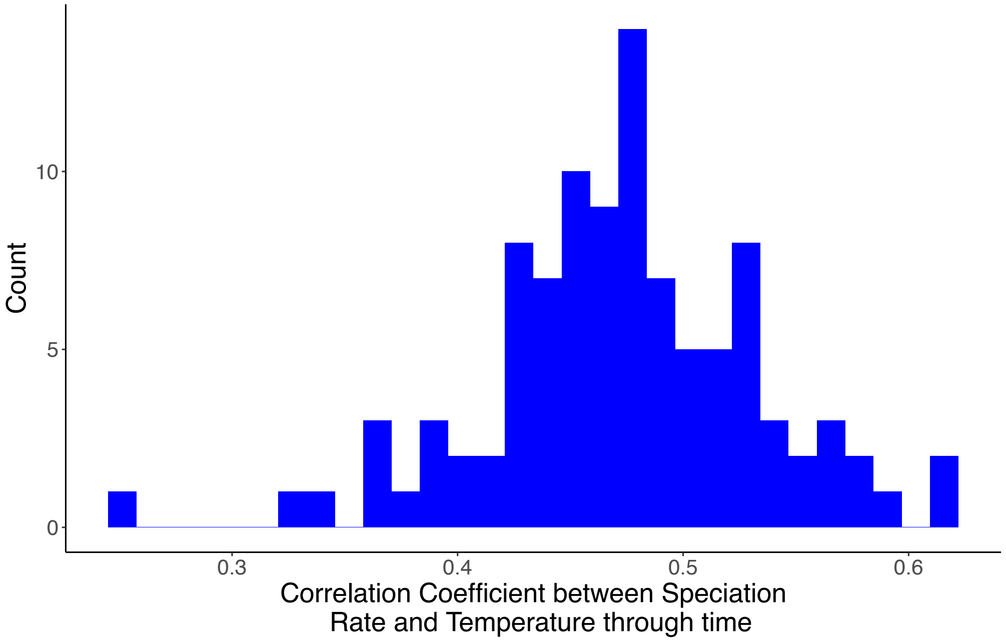


Figure 9: Correlation coefficient between marine Pseudosuchia speciation rate and temperature through time. *Histogram showing the distribution of the marine Pseudosuchia speciation rate and temperature through time correlation coefficients (mean = 0.472 ± 0.0614).*

Interpretation

In accordance with Martin et al. (2014), findings reject our hypothesis. This implies higher temperature sensitivity of marine Pseudosuchia diversification rates than suggested by conservative studies (Mannion et al., 2015). The aforementioned criticized the removal of all Metriorhynchoidea species by Martin et al. (2014), (the inclusion of which resulted in no correlation). However, analysed data here included Metriorhynchoidea species, and obtained significant, positive correlation.

4. Macroevolutionary time scale extinctions of marine Pseudosuchia and temperature

Hypothesis

There will be no correlation between marine Pseudosuchia extinction rates and global temperature.

Statistical testing

To test this hypothesis, the correlation coefficient between a time series of global temperatures (**Figure 1**) and an extinction rate time series of marine Pseudosuchia (**Figure 10**) (modelled from phylogeny (**Figure 8**)) was calculated.***

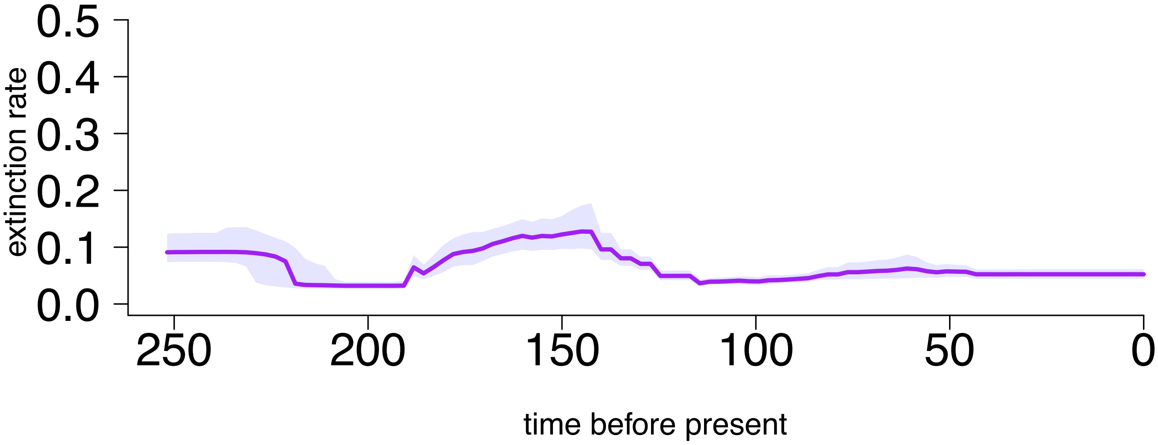


Figure 10: Extinction rate of marine Pseudosuchia through time. *Marine Pseudosuchia extinction rate (modelled from the group's phylogeny and presented in species per million years) plotted against geological time (in millions of years) from ~252 million years ago, to present time*. The solid purple line and light blue shaded area correspond to the speciation rate through time and confidence intervals, respectively. [*Plot is displaced by ~6.29 million years due to an issue with BAMMtools (Rabosky et al., 2014) making it impossible to adjust in R studio. Times presented in the plot should be read as ~6.29 units larger, making the time frame ~258 million years ago to ~6.29 million years ago]. Data from Payne et al. (In prep).*

Result

There was no correlation (r=0.0226±0.0538) between marine Pseudosuchia extinction rates and temperature (Wilcoxon unpaired test: p = 0.0007; 95% CI[-0.0566, 0.141)(**Figure 11**).

[redacted]

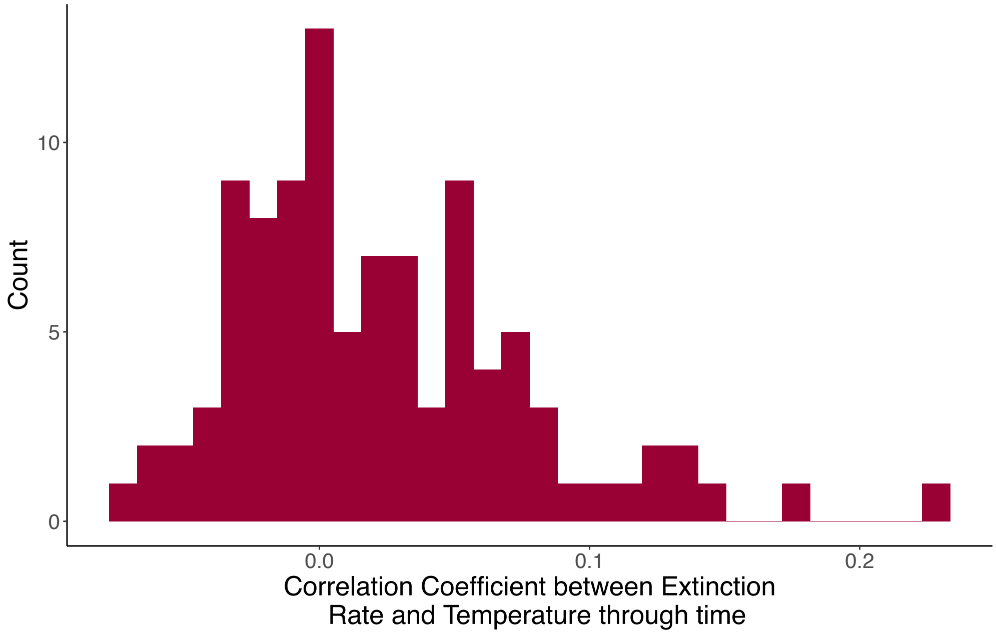


Figure 11: Correlation coefficient between marine Pseudosuchia extinction rate and temperature through time. *Histogram showing the distribution of the marine Pseudosuchia extinction rate and temperature through time correlation coefficients (mean = 0.0226 ± 0.0538).*

Interpretation

The hypothesis is accepted, and supported by previous research (Mannion et al., 2015). As with terrestrial taxa, extinction in Pseudosuchia could be less temperature dependent than expected, (Martin et al., 2014). However, this is the more accepted scenario, unlike terrestrial temperature-independent extinction. Other abiotic factors could be responsible for these dynamics.

5. Sea-level rise (on macroevolutionary time-scales) decreases terrestrial Pseudosuchia speciation

Hypothesis

Speciation rates in terrestrial Pseudosuchia and global sea-level will be significantly negatively correlated.

Statistical testing

To test this hypothesis, the correlation coefficient between a time series of eustatic sea level(**Figure 12**) and a speciation rate time series of terrestrial Pseudosuchia(**Figure 2**) was calculated.***

[redacted]

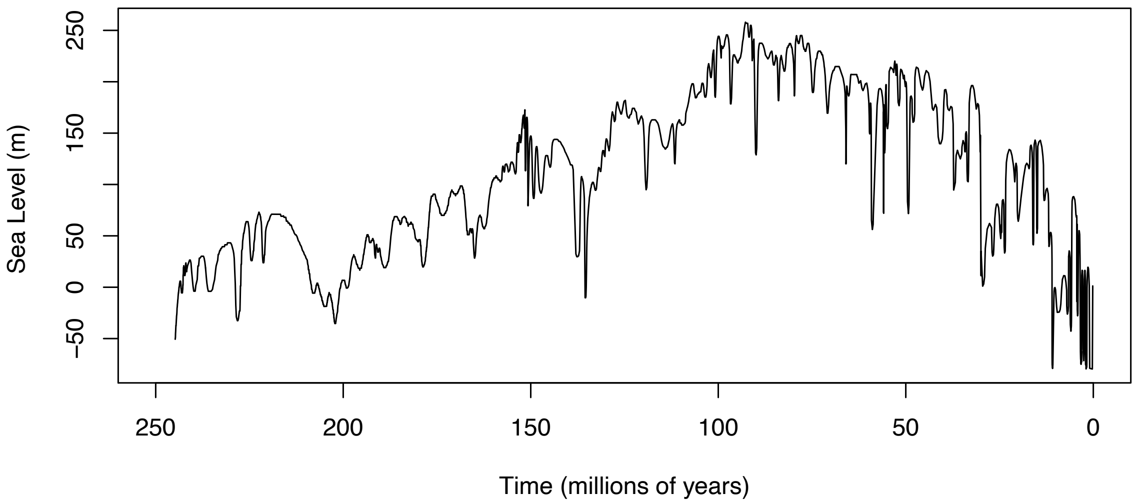


Figure 12: Global (eustatic) sea level through geological time. *Global eustatic sea level change from paleoenvironmental proxies (sedimentology) presented in meters and plotted against time (in millions of years) from ~245 million years ago, to present. Data from Haq, Hardenbol and Vail (1987).*

Result

There was a significant negative correlation (r=-0.434±0.117) between terrestrial Pseudosuchia speciation rate and sea-level(Wilcoxon unpaired test: p=3.96x10-18; 95% CI[-0.625, -0.143])(**Figure 13**).

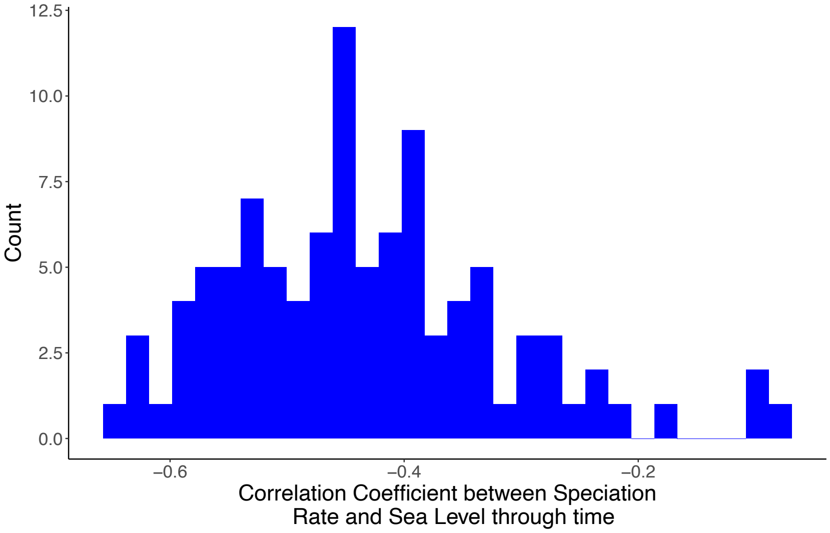


Figure 13: Correlation coefficient between terrestrial Pseudosuchia speciation rate and global (eustatic) sea level through time. *Histogram showing the distribution of the terrestrial Pseudosuchia speciation rate and sea level through time correlation coefficients (mean = -0.434 ± 0.117).*

[redacted]

Interpretation

These findings, in syntony with previous studies (Klausen, Paterson and Benton, 2020), suggest acceptance of the hypothesis. The inverse relationship between global sea-level and terrestrial Pseudosuchia speciation is potentially due to reduced coastal habitats and associated ecosystem impacts brought about by sea-level rises. Sea-level effects on other pseudosuchian diversification components (extinction) and ecological groups should be explored.

6. Marine Pseudosuchia Macroevolutionary time scale speciation and sea-level

Hypothesis

Speciation rates in marine Pseudosuchia and global sea-level will be significantly positively correlated.

Statistical testing

To test this hypothesis, the correlation coefficient between a time series of eustatic sea-level(**Figure 12**) and a speciation rate time series of marine Pseudosuchia(**Figure 7**) was calculated.***

Result

There was no correlation (r=-0.0588±0.124) between marine Pseudosuchia speciation rates and sea-level (Wilcoxon unpaired test:p=2.75x10-5; 95% CI[-0.345, 0.147])(**Figure 14**).

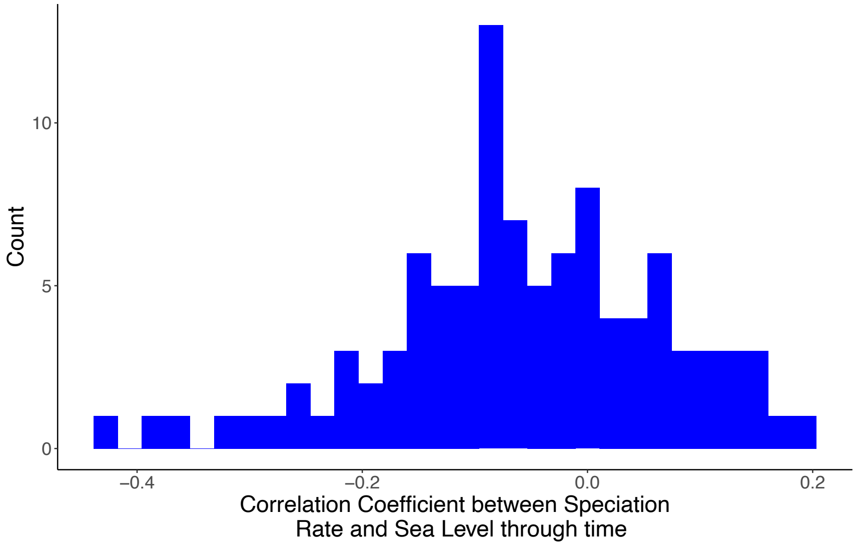


Figure 14: Correlation coefficient between marinel Pseudosuchia speciation rate and global (eustatic) sea level through time. *Histogram showing the distribution of the marine Pseudosuchia speciation rate and global sea level through time correlation coefficients (mean = -0.0588 ± 0.124).*

Interpretation

Findings are not in line with existing literature and suggest rejecting the hypothesis. There was ample evidence for significant positive correlations between sea-level and marine Pseudosuchia speciation rate, explained by habitat expansion, and increased marine shelf biodiversity (including food sources) (Mannion et al., 2015; Tennant, Mannion and Upchurch, 2016). These results are contentious, sources of error should be considered.

7. Terrestrial Pseudosuchia Macroevolutionary time scale extinctions and sea-level

Hypothesis

Extinction rates in terrestrial Pseudosuchia and global sea-level will be significantly positively correlated.

Statistical testing

To test this hypothesis, the correlation coefficient between a time series of eustatic sea-level (**Figure 12**) and an extinction rate time series of terrestrial Pseudosuchia(**Figure 5**) was calculated.***

Result

There was a significant positive correlation (r=0.306±0.142) between terrestrial Pseudosuchia extinction rates and sea-level (Wilcoxon unpaired test:p=1.13x10-17; 95% CI[-0.0427, 0.539])(**Figure 15**).

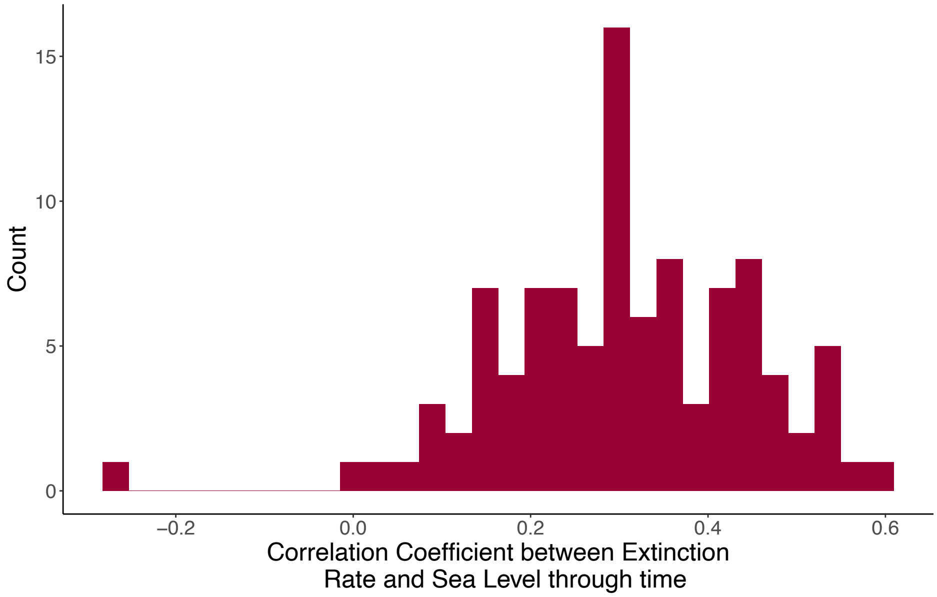


Figure 15: Correlation coefficient between terrestrial Pseudosuchia extinction rate and global (eustatic) sea level through time*. Histogram showing the distribution of the terrestrial Pseudosuchia extinction rate and sea level through time correlation coefficients (mean = 0.306 ± 0.142).*

Interpretation

In line with previous studies, the hypothesis is accepted. As mentioned regarding speciation rates, rising sea-levels have been found to restrict coastal habitats of these reptiles, and significantly impact their ecosystems, possibly causing extinctions, and hampering speciation (Klausen, Paterson and Benton, 2020).

8. Marine Pseudosuchia Macroevolutionary time scale extinctions and sea-level

Hypothesis

Extinction rates of marine Pseudosuchia and global sea-level will be significantly negatively correlated.

Statistical testing

To test this hypothesis, the correlation coefficient between a time series of eustatic sea-level (**Figure 12**) and an extinction rate time series of marine Pseudosuchia (**Figure 10**) was calculated using a DCCA-based test. The significance of this correlation was calculated using a Wilcoxon unpaired test. ***All correlations were calculated using a DCCA-based test, with significance calculated using a Wilcoxon unpaired test, and all calculations carried out using R studio (Bauer, 1972; Hollander and Wolfe, 1973; Wickham, 2016; RStudio Team, 2021).***

Result

There was no correlation (r=0.0605±0.108) between marine Pseudosuchia extinction rates and sea-level (Wilcoxon unpaired test: p=3.91x10-7; 95% CI[-0.0887, 0.287])(**Figure 16**).

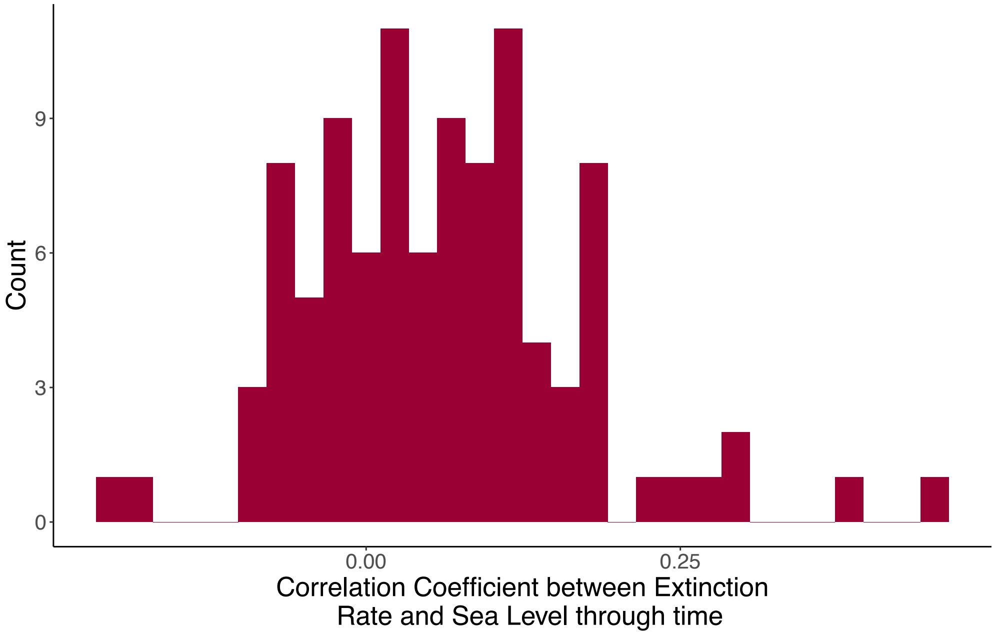


Figure 16: Correlation coefficient between marine Pseudosuchia extinction rate and global (eustatic) sea level through time. *Histogram showing the distribution of the marine Pseudosuchia extinction rate and global sea level through time correlation coefficients (mean = 0.0605 ± 0.108).*

Interpretation

Contrary to previous findings (Mannion et al., 2015; Tennant, Mannion and Upchurch, 2016), present ones suggest rejection of the hypothesis. Previous conclusions make sense (e.g. lower sea-level resulting in habitat loss and ecosystem change, leading to a decline in biodiversity and extinctions (IBID)). Nevertheless, discrepancy source could be the approach for calculation of extinction-sea-level correlations. Previous work focused on specific major biodiversity declines (Tennant, Mannion and Upchurch, 2016), whereas here the scale was macroevolutionary. Maybe at this scale, sea-level effects were too punctuated to be noticed. Again, bias in the fossil record should be considered. There could be other more important drivers of extinction within this group.

[redacted]

CONCLUSION

Agreeing with the literature, global temperature increased terrestrial speciation and did not affect marine extinction rates; and global sea-level rise negatively and positively correlated with terrestrial extinction and speciation, respectively. A lack of correlation between temperature and terrestrial extinctions, and sea-level and marine speciation were controversial, suggesting fossil bias, or lower temperature effects for the former. Interestingly, no correlations between marine extinction and sea-level and positive marine speciation-temperature correlation were found. Ectothermic physiology likely underlies temperature's effects on Pseudosuchia, and habitat restrictions/expansions underly sea-level effects. Without excluding the possibility for other factors having significant effects, these environmental parameters play a key role that is just beginning to be unravelled.

REFERENCES

Bauer, D. F. (1972) ‘Constructing Confidence Sets Using Rank Statistics’, Journal of the American Statistical Association, 67(339), pp. 687–690.

Bellard, C. et al. (2012) ‘Impacts of climate change on the future of biodiversity’, Ecology letters, 15(4), pp. 365–377.

Benton, M. J. (2009) ‘The Red Queen and the Court Jester: species diversity and the role of biotic and abiotic factors through time’, Science, 323(5915), pp. 728–732.

Carvalho, I. de S. et al. (2010) ‘Climate’s role in the distribution of the Cretaceous terrestrial Crocodyliformes throughout Gondwana’, Palaeogeography, palaeoclimatology, palaeoecology, 297(2), pp. 252–262.

Cubo, J. et al. (2020) ‘Were Notosuchia (Pseudosuchia: Crocodylomorpha) warm-blooded? A palaeohistological analysis suggests ectothermy’, Biological journal of the Linnean Society. Linnean Society of London, 131(1), pp. 154–162.

De Celis, A., Narváez, I. and Ortega, F. (2020) ‘Spatiotemporal palaeodiversity patterns of modern crocodiles (Crocodyliformes: Eusuchia)’, Zoological journal of the Linnean Society, 189(2), pp. 635–656.

Dunne, E. M. et al. (2021) ‘Climatic drivers of latitudinal variation in Late Triassic tetrapod diversity’, Palaeontology, 64(1), pp. 101–117.

Haq, B. U., Hardenbol, J. and Vail, P. R. (1987) ‘Chronology of fluctuating sea levels since the triassic’, Science, 235(4793), pp. 1156–1167.

Hollander, M. and Wolfe, D. A. (1973) Nonparametric Statistical Methods. John Wiley & Sons.

iucncsg.org (2021). Available at: http://www.iucncsg.org/ (Accessed: 26 April 2021).

Klausen, T. G., Paterson, N. W. and Benton, M. J. (2020) ‘Geological control on dinosaurs’ rise to dominance: Late Triassic ecosystem stress by relative sea level change’, Terra nova, 32(6), pp. 434–441.

Legendre, L. (2014) Did crocodiles become secondarily ectothermic ? : a paleohistological approach. Université Pierre et Marie Curie - Paris VI. Available at: https://tel.archives-ouvertes.fr/tel-01205158/document (Accessed: 25 April 2021).

Legendre, L. J. et al. (2016) ‘Palaeohistological Evidence for Ancestral High Metabolic Rate in Archosaurs’, Systematic biology, 65(6), pp. 989–996.

Mannion, P. D. et al. (2015) ‘Climate constrains the evolutionary history and biodiversity of crocodylians’, Nature communications, 6, p. 8438.

Markwick, P. J. (1998) ‘Crocodilian Diversity in Space and Time: The Role of Climate in Paleoecology and its Implication for Understanding K/T Extinctions’, Paleobiology, 24(4), pp. 470–497.

Martin, J. E. et al. (2014) ‘Sea surface temperature contributes to marine crocodylomorph evolution’, Nature communications, 5, p. 4658.

Payne, A. R. D. et al. (no date) ‘Decoupling speciation and extinction reveals both abiotic and biotic drivers shaped 250 million years of diversity on crocodile-line archosaurs’.

Pereira, H. M., Navarro, L. M. and Martins, I. S. (2012) ‘Global Biodiversity Change: The Bad, the Good, and the Unknown’, Annual review of environment and resources, 37(1), pp. 25–50.

Rabosky, D. L. et al. (2014) ‘BAMMtools: an R package for the analysis of evolutionary dynamics on phylogenetic trees’, Methods in ecology and evolution / British Ecological Society, 5, pp. 701–707.

de Ricqlès, A. J., Padian, K. and Horner, J. R. (2003) ‘On the bone histology of some Triassic pseudosuchian archosaurs and related taxa’, Annales de Paléontologie, 89(2), pp. 67–101.

Rosenzweig, C. et al. (2008) ‘Attributing physical and biological impacts to anthropogenic climate change’, Nature, 453(7193), pp. 353–357.

RStudio Team (2021) ‘RStudio: Integrated Development Environment for R’. Boston, MA. Available at: http://www.rstudio.com/.

Shirley, M. H. and Austin, J. D. (2017) ‘Did Late Pleistocene climate change result in parallel genetic structure and demographic bottlenecks in sympatric Central African crocodiles, Mecistops and Osteolaemus?’, Molecular ecology, 26(22), pp. 6463–6477.

Tennant, J. P., Mannion, P. D. and Upchurch, P. (2016) ‘Environmental drivers of crocodyliform extinction across the Jurassic/Cretaceous transition’, Proceedings. Biological sciences / The Royal Society, 283(1826), p. 20152840.

Veizer, J. et al. (1999) ‘87Sr/86Sr, δ13C and δ18O evolution of Phanerozoic seawater’, Chemical geology, 161(1), pp. 59–88.

Wickham, H. (2016) ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.

Young, M. T. et al. (2010) ‘The evolution of Metriorhynchoidea (mesoeucrocodylia, thalattosuchia): an integrated approach using geometric morphometrics, analysis of disparity, and biomechanics’, Zoological journal of the Linnean Society, 158(4), pp. 801–859.

Zachos, J. et al. (2001) ‘Trends, rhythms, and aberrations in global climate 65 Ma to present’, Science, 292(5517), pp. 686–693.

[redacted]

SUPPLEMENTARY METHODS

```
##############################################################
# 1.  Loading, exploring and making plots with the environmental data    #
#               (Sea level and Temperature vs Time Plots)                #
##############################################################

#Here, we're just using the environmental data
#(i.e. temperatureTimeSeries.csv and seaLevelTimeSeries.csv)

#setting working directory
setwd("~/Desktop/Y2 York Bio/10 Big data Biology/spring term (start)/4. Workshop 1 (pseudosucchia data set- Katie Davis) ")

#TEMPERATURE DATA SET = time in millions of years (left column)
#and temperature *proxy* (*not actual temperatures*) (right column)
#= time series showing us how temperature
#has changed over millions of years

#reading in the temperature data
temperature <- read.csv("data (pseudosucchia)/temperatureTimeSeries.csv", header= FALSE)
```

[redacted]

```
)#to see it in console window
head(temperature)

#to know what kind of data it is
class(temperature)
#its a data.frame

#to check dimensions
dim(temperature)

#we can use $ to extract elements from our
#data frame
temperature$V1
temperature$V2

#PLOTTING it
plot(temperature)
#default= open circles (kind of a scatterplot)
#also noisy, with lot's of points overlapping each other,
#which can cause problems with autocorrelation, so we
#do need to smooth the curve to get rid of some of this

#SMOOTHING THE CURVE
#(Autocorrelation is when a time series correlates
#with itself, and this can happen when
#you have repetitive patterns in time series and
#even though we're going to be correlating the temperature
#time series with the speciation rate time series
#in the later workshop, the analysis could still
#pick up on any autocorrelation within the temperature
#time series, and this can give an erroneous result;
# + we do know that temperature can be cyclical over
#geological time scales, so we do need to smooth the
#curve to try to remove autocorrelation).

finaltemp <- smooth(smooth(temperature$V2))
#we're smoothing it twice because this is what seems
#to give the best results

#PLOT FINAL TEMP
plot(finaltemp)

#the x axis on this is wrong, because we get an index instead of time,
#and this is because we did not ask R to plot temperature against time...
```

[redacted]

```
#Plotting temperature against time
plot(temperature$V1, finaltemp,xlab = "Time (millions of years)", ylab = "Temperature",
    type = "l")
#V1 is time, and finaltemp is y
#but, the time in the x-axis starts at 0 and goes back in time towards the right
# = not intuitive

#reversing the direction of the x-axis
plot(temperature$V1, finaltemp,xlab = "Time (millions of years)", ylab = "Temperature",
    xlim =c(220,0))
#done! we start at 220 because that is what our time series
#goes up to- 220 million years ago

#PLOTTING A "BLACK LINE"
#(to make the plot look nicer, using type = "l" turns this into a
#"line plot" instead of what looks like a scatterplot)
plot(temperature$V1, finaltemp,xlab = "Time (millions of years)", ylab = "Temperature",
    xlim =c(220,0), type="l", cex.lab=1.56)
#Added code:
#cex.lab to make the axis labels bigger

####UP TO HERE to produce a plot of temperature and time (saved as "Temperature vs. time
(mya).pdf")######################################################################

#SECOND SET OF ENVIRONMENTAL DATA
#SEA LEVEL DATA OVER GEOLOGICAL TIME
#1st column = time, labeled as age (millions of years)
#2nd column = sealevel, labeled as SL (meters)
# ^sealevel seen here is what is known as eustatic sealevel
#which means that it measures a global alteration to the
#volume of water in the oceans- so while the temperature
#curve was global temperature, this is global sealevel,
#so we're looking at an entire globe climate change.

#LOADING OUR SEALEVEL DATA INTO R
seaLevel <- read.csv("data (pseudosucchia)/seaLevelTimeSeries.csv", header= TRUE)

#to take a look at it in our console
head(seaLevel)
#checking data set was imported correctly
class(seaLevel)
dim(seaLevel)
```

[redacted]

```
#to look at single elements using $
seaLevel$Age
seaLevel$SL

#PLOTTING SEALEVEL
plot(seaLevel)
#sea level measured in meters on the y-axis and time measured
#in millions of years on the x-axis

#data is not too noisy and the axes are "sensible", so we
#can skip the first few steps we did on the temperature
#data, but label the axes again

plot(seaLevel, xlab = "Time (millions of years)",
    ylab =  "Sea Level (m) " )

#to reverse the x-axis
plot(seaLevel, xlab = "Time (millions of years)",
    ylab =  "Sea Level (m) ", xlim = c(250,0))

#PLOTING A LINE (to make the data look nicer)
plot(seaLevel, xlab = "Time (millions of years)",
    ylab =  "Sea Level (m) ", xlim = c(250,0),
    type = "l")
##UP TO HERE TO PRODUCE THE SEA LEVEL PLOT SEEN IN THE REPORT####################

#Saving the data produced so far
save.image("Workshop 1 (pseudosucchia).RData")

##################################################################
#      2.                 Exploring phylogenetic data and                        #
#                            plotting phylogenetic trees                          #
#              (Terrestrial and Marine Pseudosuchia Phylogenies)                #
##################################################################

#data sets to be used: phylogenetic (fossilCrocPhylogeny.tre) and habitat (HabitatData.csv)
#loading libraries
library(phytools)
library(strap)

#loading in the phylogeny using the read.tree function, and putting the tree into a
#variable called "tree"
tree <- read.tree("data (pseudosucchia) copy 1 /fossilCrocPhylogeny.tre")
```

[redacted]

```
#taking a look at our tree by plotting it
plot(tree)

#the plot is not really helpful (it is too big for us to be able
#to see anything)

#To take a look at what is in the tree in another way:
#we take a look at the tree object
tree

#Output
#Phylogenetic tree with 536 tips and 535 internal nodes.
#
#Tip labels:
#  Acaenasuchus_geoffreyi, Desmatosuchus_haplocerus, Desmatosuchus_smalli,
Lucasuchus_hunti, Sierritasuchus_macalpini, Longosuchus_meadei, ...
#
#Rooted; includes branch lengths.

#tree tips= species or OTUs (operational taxonomic units)
#node= point at which two branches join
#root= the deepest (oldest) node in the tree
#the number of nodes is 535 because it is a fully bifurcated tree-
#always containing one fewer node than the number of
#taxa it contains

#the tree is rooted = it shows ancestral relationships
#unrooted trees only show the relative relatedness of
#the taxa in the tree- by definition, a phylogeny that is scaled
#to geological time will always be a rooted tree

#and, our tree contains branch lengths- the tree is scaled
#to geological time, and that means the units of branch
#lengths are millions of years

#Another way to see the tree tips (OTUs/operational
#taxonomic units)
Ntip(tree)
#[1] 536

class(tree)
#data class is a phylo object, which is what we would expect

#taking a closer look at what makes up a phylogenetic tree
```

[redacted]

```
tree$edge #branches, horizontal lines that connect nodes
tree$Nnode #points that the branches or edges connect to
tree$tip.label #OTUs or the taxa in your tree
tree$root.edge #oldest or deepest node in the tree, and root.edge is the
#branch that leads to the root, and in our tree, there is no root.edge, so if we
#run this command we get:
#0
#and this is logical because we don't have root.edge in our tree, as it is
#rooted in geological time instead of an outgroup

#using head(tree$) to see the first few contents of the tree elements
head(tree$Nnode)
#535
#= the number of nodes

head(tree$tip.label)
#[1] "Acaenasuchus_geoffreyi"   "Desmatosuchus_haplocerus" "Desmatosuchus_smalli"
#[4] "Lucasuchus_hunti"         "Sierritasuchus_macalpini" "Longosuchus_meadei"
#returns taxa in the tree

head(tree$edge.length)
#6.145823 3.724195 2.961445 2.318935 6.864920 4.590897
#these are the branches' lengths, so the first branch has a length of
#6.145823 million years

head(tree$edge)
#     [,1] [,2]
#[1,]  537  538
#[2,]  538  539
#[3,]  539  540
#[4,]  540  541
#[5,]  541  542
#[6,]  542  543

#shows us that branch #1 connects node numbers 537 and 538, and we know from
#the previous command that this edge/branch connecting these two nodes is 6.145823my
#in length

head(tree$root.edge)
#0
#again, logical, as there is no root.edge

#^introduction to the basics of what makes up a phylogenetic tree,
#now let's move on to our habitat data, so we can start to explore our
```

[redacted]

```
#tree as two partitions.

#reading in the data (.csv file)
habitatdata <- read.csv("data (pseudosucchia) copy 1 /HabitatData.csv",
                        header = T, stringsAsFactors = FALSE)

#you can check habitat data using head(habitatdata)
head(habitatdata)

#                           Taxon     Habitat
#1         Acaenasuchus_geoffreyi Terrestrial
#2     Adamanasuchus_eisenhardtae Terrestrial
#3         Adamantinasuchus_navae Terrestrial
#4             Adzhosuchus_fuscus Terrestrial
#5               Aeolodon_priscus      Marine
#6 Aetobarbakinoides_brasiliensis Terrestrial
#looks perfect!

#PLOTTING PHYLOGENY FOR THE TERRESTRIAL TAXA

#let's extract a list of terrestrial taxa using subset
TerrestrialTaxa<- subset(habitatdata, habitatdata$Habitat=='Terrestrial')$Taxon
#here we are telling R to look inside our habitat data, and asking it to
#identify anything in that data in the Habitat column that is listed as Terrestrial,
#and we only want it to return the taxon labels, not the full row (explaining $Taxon
#part which goes with the Taxon column in our data)

#now let’s take a look at our list of terrestrial taxa
head(TerrestrialTaxa)
#[1] "Acaenasuchus_geoffreyi"         "Adamanasuchus_eisenhardtae"
#[3] "Adamantinasuchus_navae"         "Adzhosuchus_fuscus"
#[5] "Aetobarbakinoides_brasiliensis" "Aetosauroides_scagliai"

#now we want to use this list to extract the phylogeny that is only made up of terrestrial
#taxa, and we call this a subtree, which we do using the function
#keep.tip which keeps the tips in our list and drops everything else
treeT <- keep.tip(tree, TerrestrialTaxa)

#taking a look at the subtree
treeT
#Phylogenetic tree with 207 tips and 206 internal nodes.
#
#Tip labels:
```

[redacted]

```
# Acaenasuchus_geoffreyi, Desmatosuchus_haplocerus, Desmatosuchus_smalli,
Lucasuchus_hunti, Sierritasuchus_macalpini, Longosuchus_meadei, ...
#
#Rooted; includes branch lengths.

#only 207 taxa and 206 nodes, and the taxa should match those in the csv file coded
#as terrestrial

#Checking the first terrestrial species on the list actually lived on
#land (Acaenasuchus_geoffreyi)
#checked and it IS!

#you can also explore the data using head(treeT$) or treeT$
treeT$edge

#PLOTTING THE TREE
plot(treeT)
#its a bit less messy than the original one, but we still can't see what's going on
#let's try to tidy up a bit, first, by changing the font size
plot(treeT, cex=0.2)
#this is a lot tidier but it is still hard to read on a small screen...

#writing the phylogenetic tree to file
write.tree(treeT, file = 'TerrestrialTaxa.tre')
#done

#PLOTTING THE TERRESTRIAL TAXA TREE SCALED TO GEOLOGICAL TIME
#this is a bit fiddly because the package we're going to use to do this wasn't
#designed for phylogenies that are already time-calibrated-this means we have to create
#an empty data frame (because that's what the package expects)

#the main thing that we need to note is the age of our root node - the node at the very base
#of the tree that everything else subtends from- the oldest node in the tree... (it
#doesn't look like a node because it has no branch leading to it but it is one and R
#knows this)

#the reason we need to know the age value of our root node is so that the strap
#library, that we're about to use, knows where to place the tree in geological time.
#so we start by grabbing the ages of all the nodes, using nodeHeights, and plot that into
#something we call lengths
lengthsT <- nodeHeights(treeT)

#taking a look inside lengths
head(lengthsT)
```

[redacted]

```
#        [,1]     [,2]
#[1,]  0.000000  6.145823
#[2,]  6.145823  9.870018
#[3,]  9.870018 12.831463
#[4,] 12.831463 15.150398
#[5,] 15.150398 22.015318
#[6,] 22.015318 26.606215
#this is showing the node ages in the columns labelled 1 and 2 as measured
#from the root for each edge/branch.

#the reason we need to do this is because we need to work out which is the biggest
#number/node age (=oldest node=root) as this will give us our root age that we need to plot a
tree.

#The assumptions made here to work out the root age using this data set are that there are some
extant taxa…

#Now that we have all the ages, we can work out the maximum,
#using maxlengths
root.timeT <- max(lengthsT)

#if you take a look at root.time you should now have the root age
root.timeT
#281.9737

#Strap expects a variable called root.time inside the tree object, so we're going to
#create that now using the root time that we have just calculated
treeT$root.time <- root.timeT

#now we're going to create out empty data frame- remember this will be empty
#because we're not actually going to use it, the strap expects it so we have to provide it anyway.
#our plot will actually use the dates from our phylogeny
#our empty data frame needs two columns labelled FAD and LAD, with each row corresponding
#to an OTU- FAD and LAD refer to the earliest and latest dates that a taxon is known to
#occur in the fossil record (FAD = first appearance datum; LAD= Last appearance datum)

#so, first we get a list of all of the tip labels
all_otusT <-treeT$tip.label

#having a look to see if it looks right
head(all_otusT)
#[1]    "Acaenasuchus_geoffreyi"         "Desmatosuchus_haplocerus"    "Desmatosuchus_smalli"
"Lucasuchus_hunti"
#[5] "Sierritasuchus_macalpini" "Longosuchus_meadei"
```

```
#looks good

#now we create our empty matrix
all_otudatesT <-matrix(0,nrow = length(all_otusT), ncol = 2)
#here, we're telling R to set the rows to the otus, we're telling it that we need two
#columns (one for fad and one for lad) and we're telling it to populate the matrix with 0s

#taking a look at it
head(all_otudatesT)
#    [,1] [,2]
#[1,]    0    0
#[2,]    0    0
#[3,]    0    0
#[4,]    0    0
#[5,]    0    0
#[6,]    0    0
#looks good

#now we need to convert our matrix into a data frame, as a strap requires it
all_otudatesT <- data.frame(all_otudatesT)

#having a look at it
head(all_otudatesT)
#  X1 X2
#1  0  0
#2  0  0
#3  0  0
#4  0  0
#5  0  0
#6  0  0
#looks good

#now we need to label the rows with the otus rather than just a number
row.names(all_otudatesT) <- all_otusT
#checking
head(all_otudatesT)
#                       X1 X2
#Acaenasuchus_geoffreyi    0  0
#Desmatosuchus_haplocerus  0  0
#Desmatosuchus_smalli      0  0
#Lucasuchus_hunti          0  0
#Sierritasuchus_macalpini  0  0
#Longosuchus_meadei        0  0
#looks good, we now have our taxon labels as the row names
```

[redacted]

```
#the last thing we want to do is change the column names from X1 and X2 to
#FAD and LAD
colnames(all_otudatesT) <- c('FAD', 'LAD')
#checking
head(all_otudatesT)
#                       FAD LAD
#Acaenasuchus_geoffreyi     0   0
#Desmatosuchus_haplocerus   0   0
#Desmatosuchus_smalli       0   0
#Lucasuchus_hunti           0   0
#Sierritasuchus_macalpini   0   0
#Longosuchus_meadei         0   0
#perfect!
#rows labelled as your taxa/OTUs and two columns labelled FAD and LAD

#Now, we can plot our tree against the geological timescale
treeTplotw_geologicalperiodsandepochs<-geoscalePhylo(treeT,
            ages = all_otudatesT,
            cex.tip = 0.1,
            lwd=1,
            quat.rm=T,
            units= c("Period", "Epoch"),
            boxes="Epoch")

treeTplotw_geologicalperiodsandepochs
#amazing

#UP TO HERE FOR THE TERRESTRIAL PSEUDOSUCHIA PHYLOGENY PLOT###################

#Saving the data we have so far
save.image("Crocs_Wrokshop2.RData")

#PLOTTING PHYLOGENY FOR THE MARINE TAXA

#let's extract a list of *marine* taxa using subset,
#and call the subset MarineTaxa
MarineTaxa<- subset(habitatdata, habitatdata$Habitat=='Marine')$Taxon
#here we are telling R to look inside our habitat data, and asking it to
#identify anything in that data in the Habitat column that is listed as Marine,
#and we only want it to return the taxon labels, not the full row (explaining $Taxon
#part which goes with the Taxon column in our data)

#now let’s take a look at our list of marine taxa
```

[redacted]

```
head(MarineTaxa)
#[1] "Aeolodon_priscus"
#[2] "Aktiogavialis_caribesi"
#[3] "Aktiogavialis_puertoricensis"
#[4] "Argochampsa_krebsi"
#[5] "Atlantosuchus_coupatezi"
#[6] "cf_Terminonaris_robusta_SMNH_P2411_dot_1"

#now we want to use this list to extract the phylogeny that is only made up of *marine*
#taxa, (= subtree), using the function keep.tip which keeps the tips in our list and drops
#everything else
treeM <- keep.tip(tree, MarineTaxa)

#taking a look at the subtree
treeM
#Phylogenetic tree with 108 tips and 107 internal nodes.
#
#Tip labels:
# Atlantosuchus_coupatezi, Guarinisuchus_munizi, Rhabdognathus_acutirostris,
Rhabdognathus_aslerensis, Dyrosaurus_maghribensis, Dyrosaurus_phosphaticus, ...
#
#Rooted; includes branch lengths.

#only 108 taxa and 107 nodes, and the taxa should match those in the csv file coded
#as marine

#checking the first marine species name (Atlantosuchus_coupatezi)
#actually lived at sea (using paleobiology data base
#checked and it IS!

#you can also explore the data using head(treeM$) or treeM$
treeM$edge

#PLOTTING PHYLOGENY OF MARINE PSEUDOSUCHIA

plot(treeM)
#its a bit less messy than the original one, but we still can't see what's going on

#let's try to tidy up a bit, first, by changing the font size
plot(treeM, cex=0.2)
#this is a lot tidier but it is still hard to read on a small screen...

# writing the phylogenetic tree to file
write.tree(treeM, file = 'MarineTaxa.tre')
```

[redacted]

```
#done

#PLOTTING THE TREE SCALED TO GEOLOGICAL TIME
#the main thing that we need to note is the age of our root node - the node at the very base
#of the tree that everything else subtends from- the oldest node in the tree... (it
#doesn't look like a node because it has no branch leading to it but it is one and R
#knows this)
#remember, when you looked at tree$root.edge, the value returned was 0

#the reason we need to know the age value of our root node is so that the strap
#library, that we're about to use, knows where to place the tree in geological time.
#so we start by grabbing the ages of all the nodes, using nodeHeights, and plot that into
#something we call lengths
lengthsM <- nodeHeights(treeM)

#taking a look inside lengths
head(lengthsM)
#         [,1]      [,2]
#[1,]   0.00000  64.74032
#[2,]  64.74032  67.29545
#[3,]  67.29545  96.10027
#[4,]  96.10027 137.29059
#[5,] 137.29059 148.03486
#[6,] 148.03486 153.30037
#this is showing the node ages in the columns labelled 1 and 2 as measured
#from the root for each edge/branch.

#the reason we need to do this is because we need to work out which is the biggest
#number/node age (=oldest node=root) as this will give us our root age that we need to plot a
tree.

#moving on with our plot, now that we have all the ages, we can work out the maximum,
#using maxlengths
root.timeM <- max(lengthsM)

#if you take a look at root.time you should now have the root age
root.timeM
#[1] 251.8147

#Strap expects a variable called root.time inside the tree object, so we're going to
#create that now using the root time that we have just calculated
treeM$root.time <- root.timeM

#now we're going to create the previously mentioned empty data frame
```

```
#that strap expects.

#first, we get a list of all of the tip labels
all_otusM <-treeM$tip.label

#having a look to see if it looks right
head(all_otusM)
#[1] "Atlantosuchus_coupatezi"    "Guarinisuchus_munizi"
#[3] "Rhabdognathus_acutirostris" "Rhabdognathus_aslerensis"
#[5] "Dyrosaurus_maghribensis"    "Dyrosaurus_phosphaticus"
#looks good

#now we create our empty matrix
all_otudatesM <-matrix(0,nrow = length(all_otusM), ncol = 2)
#here, we're telling R to set the rows to the otus, we're telling it that we need two
#columns (one for fad and one for lad) and we're telling it to populate the matrix with 0s

#taking a look at it
head(all_otudatesM)
#     [,1] [,2]
#[1,]    0    0
#[2,]    0    0
#[3,]    0    0
#[4,]    0    0
#[5,]    0    0
#[6,]    0    0
#looks good

#now we need to convert our matrix into a data frame, as a strap requires it
all_otudatesM <- data.frame(all_otudatesM)
#having a look at it
head(all_otudatesM)
#  X1 X2
#1  0  0
#2  0  0
#3  0  0
#4  0  0
#5  0  0
#6  0  0
#looks good

#now we need to label the rows with the otus rather than just a number
row.names(all_otudatesM) <- all_otusM
#checking
```

```
head(all_otudatesM)
#                        X1 X2
#Atlantosuchus_coupatezi     0  0
#Guarinisuchus_munizi       0  0
#Rhabdognathus_acutirostris  0  0
#Rhabdognathus_aslerensis    0  0
#Dyrosaurus_maghribensis     0  0
#Dyrosaurus_phosphaticus     0  0
#looks good, we now have our taxon labels as the row names

#the last thing we want to do is change the column names from X1 and X2 to
#FAD and LAD
colnames(all_otudatesM) <- c('FAD', 'LAD')
#checking
head(all_otudatesM)
#                        FAD LAD
#Atlantosuchus_coupatezi      0   0
#Guarinisuchus_munizi        0   0
#Rhabdognathus_acutirostris   0   0
#Rhabdognathus_aslerensis     0   0
#Dyrosaurus_maghribensis      0   0
#Dyrosaurus_phosphaticus      0   0
#perfect!
#rows labelled as your taxa/OTUs and two columns labelled FAD and LAD

#Now, we can plot our tree against the geological timescale
treeMplotw_geologicalperiodsandepochs<-geoscalePhylo(treeM,ages=all_otudatesM,
cex.tip=0.1, lwd=1, quat.rm=T, units=c("Period", "Epoch"), boxes="Epoch")

#taking a look at it
treeMplotw_geologicalperiodsandepochs
```

#BUT, THERE IS A PROBLEM WITH THE RESULTING
#PHYLOGENETIC TREE

```
#There is one species who's branch reaches the present
#(Piscogavialis jugaliperforatus),
#which is quite odd as there are no extant marine Pseudosuchia.

#this error could have arisen due to the root age calculation assuming
#there are extant taxa within marine Pseudosuchia, when there are none.

#To check if this is the source of the error, the unpartitioned tree
#can be plotted and to see if the branch of P. jugaliperforatus still
```

[redacted]

```
#reaches the present time.

##################################################################################
#  3.        Plotting an unpartitioned Phylogenetic tree to check the extinction            #
#                          time of Piscogavialis jugaliperforatus                           #
##################################################################################

#loading libraries
library(phytools)
library(strap)

#PLOTTING A TREE WITHOUT PARTITIONING (treeA for tree "all")

#creating a variable for both terrestrial and marine taxa
#from habitat data
All_taxa<- habitatdata$Taxon

#using this list of taxa to extract the phylogeny of
#all taxa, using the function keep.tip which
#keeps the tips in our list and drops everything else
treeA <- keep.tip(tree, All_taxa)

#taking a look at the tree
treeA
#Phylogenetic tree with 315 tips and 314 internal nodes.
#
#Tip labels:
#     Acaenasuchus_geoffreyi,     Desmatosuchus_haplocerus,     Desmatosuchus_smalli,
Lucasuchus_hunti, Sierritasuchus_macalpini, Longosuchus_meadei, ...
#
#Rooted; includes branch lengths.

#grabbing the ages of all the nodes, using nodeHeights,
#and putting that into something we call lengths
lengthsA <- nodeHeights(treeA)

#now that we have all the ages, we can work out the maximum,
#using maxlengths
root.timeA <- max(lengthsA)

#if you take a look at root.timeA you should now have the root age
root.timeA

#Strap expects a variable called root.time inside the tree object, so we're going to
```

[redacted]

```
#create that now using the root time that we have just calculated
treeA$root.time <- root.timeA

#creating our empty data frame:
#first we get a list of all of the tip labels
all_otusA <-treeA$tip.label

#having a look to see if it looks right
head(all_otusA)
#[1]  "Acaenasuchus_geoffreyi"      "Desmatosuchus_haplocerus"   "Desmatosuchus_smalli"
"Lucasuchus_hunti"
#[5] "Sierritasuchus_macalpini" "Longosuchus_meadei"
#looks good

#now we create our empty matrix
all_otudatesA <-matrix(0,nrow = length(all_otusA), ncol = 2)
#here, we're telling R to set the rows to the otus, we're telling it that we need two
#columns (one for fad and one for lad) and we're telling it to populate the matrix with 0s

#taking a look at it
head(all_otudatesA)
#    [,1] [,2]
#[1,]    0    0
#[2,]    0    0
#[3,]    0    0
#[4,]    0    0
#[5,]    0    0
#[6,]    0    0
#looks good

#converting our matrix into a data frame, as strap requires it
all_otudatesA <- data.frame(all_otudatesA)

#having a look at it
head(all_otudatesA)
#  X1 X2
#1  0  0
#2  0  0
#3  0  0
#4  0  0
#5  0  0
#6  0  0
#looks good
```

[redacted]

```
#labelling the rows with the otus rather than just a number
row.names(all_otudatesA) <- all_otusA
#checking
head(all_otudatesA)
#                       X1 X2
#Acaenasuchus_geoffreyi    0  0
#Desmatosuchus_haplocerus  0  0
#Desmatosuchus_smalli      0  0
#Lucasuchus_hunti          0  0
#Sierritasuchus_macalpini  0  0
#Longosuchus_meadei        0  0
#looks good, we now have our taxon labels as the row names

#the last thing we want to do is change the column names from X1 and X2 to
#FAD and LAD
colnames(all_otudatesA) <- c('FAD', 'LAD')
#checking
head(all_otudatesA)
#                       FAD LAD
#Acaenasuchus_geoffreyi     0   0
#Desmatosuchus_haplocerus   0   0
#Desmatosuchus_smalli       0   0
#Lucasuchus_hunti           0   0
#Sierritasuchus_macalpini   0   0
#Longosuchus_meadei         0   0
#perfect!
#rows labelled as your taxa/OTUs and two columns labelled FAD and LAD

#Now, we can plot our tree against the geological timescale
treeALLplotw_geologicalperiodsandepochs<-geoscalePhylo(treeA,
                                  ages = all_otudatesA,
                                  cex.tip = 0.1,
                                  lwd=1,
                                  quat.rm=T,
                                  units= c("Period", "Epoch"),
                                  boxes="Epoch")

treeALLplotw_geologicalperiodsandepochs
#
#amazing: we get a phylogenetic tree which contains all the taxa, and in it, you
#can see that the branch leading to Piscogavialis jugaliperforatus ends ~6 million years
#ago, indicating this is the time in which it went extinct, not the present.
```

[redacted]

```
###############################################################################
#       4.          Fixing the timescale on the Phylogenetic tree plot                #
#                                   for the Marine Taxa                                #
#     (Obtaining the Marine Pseudosuchia phylogeny presented in the report)           #
###############################################################################

#In order to fix the Marine Pseudosuchia Phylogenetic tree plot,
#first we need to work out how much the marine tree is displaced by.
#This requires the package adephylo
library("adephylo")

#Checking how far away P. jugaliperforatus is from the root
distRoot(tree, tips=c('Piscogavialis_jugaliperforatus'))
#Piscogavialis_jugaliperforatus
#275.6901
#This tells us that P. jugaliperforatus is 275.6901 MYR away from the root.

#for the next part, we first need to load the required packages in
#phytools and strap libraries
library(phytools)
library(strap)

#Checking the root age of the unpartitioned tree
#Starting by grabbing the ages of all the nodes, using nodeHeights, and putting that into
#something called lengths
lengths <- nodeHeights(tree)

#now that we have all the ages, we can work out the maximum,
#using max(lengths)
root.time <- max(lengths)
#checking
root.time
#[1] 281.9796

#To know where P. jugaliperforatus should go extinct we subtract 275.6901 from 281.9796.
extinction_time_for_Piscogavialis_jugaliperforatus <- 281.9796-275.6901
extinction_time_for_Piscogavialis_jugaliperforatus
#This gives us 6.2895 MYA as the extinction time for Piscogavialis_jugaliperforatus

#adding 6.2895 to root.timeM to get the tree lined up correctly with the time scale
root.timeM <- 251.8147+6.2895
#251.8147 is the length of the longest branch in the marine taxa, so the age of the
#root, if assuming there are extant marine Pseudosuchia.
#6.895 is the newly calculated time at which Piscogavialis_jugaliperforatus
```

[redacted]

```
#must have gone extinct
treeM$root.time <- root.timeM

#checking
treeM$root.time
#[1] 258.1042
#perfect, now the root is as old as it should be in real time,
#fixed from before where it was made with the assumption that
#some species in the marine taxa were extant and dated to present time,
#which they do not (the closest one to the present is Piscogavialis_jugaliperforatus
#which as calculated should have gone extinct ~6.2895MYA)

#PLOTTING THE MARINE PHYLOGENY SCALED TO GEOLOGICAL TIME
treeMplotw_geologicalperiodsandepochs<-geoscalePhylo(treeM,
                                      ages=all_otudatesM,
                                      cex.tip=0.1, cex.ts = 0.4,
                                      lwd=1,
                                      quat.rm=T,
                                      units=c("Period", "Epoch"),
                                      boxes="Epoch")

#looking at the tree
treeMplotw_geologicalperiodsandepochs

###########UP TO HERE FOR THE MARINE PSEUDOSUCHIA PHYLOGENY #################


###############################################################################
#           5.                    Diversification rate data                   #
#     (Speciation and Extinction rate time series for terrestrial and marine Pseudosuchia)     #
###############################################################################

#checking working directory:
getwd()
#perfect
#loading previous data (from workshop 2)
load("Crocs_Workshop2.RData")

#loading libraries
library(BAMMtools)
library(phytools)

#reading in our diversification rate data and putting it into a variable called "edata"
#for eventdata
```

[redacted]

```
edata <- getEventData(tree, eventdata ='data/fossilCrocDiversificationData.txt', burnin=0.1)

#more about this data: it's called "event data" because it contains everything we
#need to explore macroevolutionary rates and events in pseudosuchia-
#we’re loading the event data for the full tree that we called "tree" previously,
#and rather than just our terrestrial subtree that we called treeT- this is because the event
#data was modelled on the whole tree and when we're loading in the data, the taxa in the tree
#have to match the taxa in the event data.
#This just means that we load the event data for the full
#tree then subset it afterwards to extract the rates for just the terrestrial subtree

#let’s take a look at what’s on the screen:
#Reading event datafile:  data/fossilCrocDiversificationData.txt
#...........
#Read a total of 10000 samples from posterior
#
#Discarded as burnin: GENERATIONS <  2997000
#Analyzing  9001  samples from posterior
#
#Setting recursive sequence on tree...
#
#Done with recursive sequence

#what does burnin mean?
#we set a burnin amount when we loaded in our edata to 0.1 or 10%
#this means that we're removing the first 10% of samples to account for the
#burn-in phase of the analysis. The rationale behind this is that the diversification
#analysis that produced these data probably didn't immediately find the best
#solutions, or, because it’s a Bayesian analysis, the most probable solutions
#removing the first 10% means that we're left with a set of results of equally
#high probability, so that's why we end up with 9001 samples from our starting number
#of 10000.
#the number of generations that we can see here is just how many iterations the analysis
#that produced this data ran for. If we had a bigger data set we would usually have
#a bigger number of generations to be sure we are finding the most optimal
#solutions.
#Before we go any further, let's go over an explanation a bit more for our
#samples (9001)
#Each one represents a separate simulation of diversification rate change through
#geological time for our tree- that means each one is a time series made up of values
#for speciation or extinction at various points through time, so
#our diversification data is made up 9001 separate time series.

#now, we can extract the rates for just our terrestrial subtree
```

[redacted]

```
#first we need to tell BAMM to extract our subtree before it can get the rates.
streeTerrestrial <- subtreeBAMM(edata, tips=TerrestrialTaxa)
#here, we're saying that we want to extract the edata associated with just the terrestrial taxa
#also, although we're calling this a subtree, it’s not quite the same as
#this subtree is a BAMM object- it contains a phylogenetic tree and its associated
#diversification rate data.

#let's take a look at our extracted subtree
streeTerrestrial
#
#Phylogenetic tree with 207 tips and 206 internal nodes.
#
#Tip labels:
# Acaenasuchus_geoffreyi, Desmatosuchus_haplocerus, Desmatosuchus_smalli,
Lucasuchus_hunti, Sierritasuchus_macalpini, Longosuchus_meadei, ...
#
#Rooted; includes branch lengths.
#
#Posterior samples: 9001
#
#List elements:
#  edge Nnode tip.label edge.length begin end downseq lastvisit numberEvents eventData
eventVectors tipStates tipLambda meanTipLambda eventBranchSegs tipMu meanTipMu type


#when comparing it to phylo object trees, some of the data are the same, both of them have
#207 tips and 206 nodes, and they
#contain the same taxa...
#but we now have some additional information in the BAMM subtree that we've just
#created: this also has our sample size ("Posterior samples: 9001", which is 9001
#because we have removed 10% as burn-in), and we also have a list of elements that
#our object contains.

#now, let's extract the rates through time for our subtree
rtt_T <- getRateThroughTimeMatrix(streeTerrestrial)

#the rates through time are speciation and extinciton rates associated with our phylogeny of
#terrestrial taxa.

#now, let's take a look at what's inside our rates through time
summary(rtt_T)
#      Length Class  Mode
#lambda 900100 -none- numeric
#mu     900100 -none- numeric
```

```
#times     100 -none- numeric
#type       1 -none- character

#this shows us we have 4 elements: lambda, mu, times, type... let's go through each
#one
#first the times
rtt_T$times
#  [1]   0.000000   2.848220   5.696439   8.544659  11.392879  14.241098  17.089318
#[8]  19.937538  22.785758  25.633977  28.482197  31.330417  34.178636  37.026856
#[15]  39.875076  42.723295  45.571515  48.419735  51.267954  54.116174  56.964394
#[22]  59.812614  62.660833  65.509053  68.357273  71.205492  74.053712  76.901932
#[29]  79.750151  82.598371  85.446591  88.294811  91.143030  93.991250  96.839470
#[36]  99.687689 102.535909 105.384129 108.232348 111.080568 113.928788 116.777007
#[43] 119.625227 122.473447 125.321667 128.169886 131.018106 133.866326 136.714545
#[50] 139.562765 142.410985 145.259204 148.107424 150.955644 153.803863 156.652083
#[57] 159.500303 162.348523 165.196742 168.044962 170.893182 173.741401 176.589621
#[64] 179.437841 182.286060 185.134280 187.982500 190.830719 193.678939 196.527159
#[71] 199.375379 202.223598 205.071818 207.920038 210.768257 213.616477 216.464697
#[78] 219.312916 222.161136 225.009356 227.857575 230.705795 233.554015 236.402235
#[85] 239.250454 242.098674 244.946894 247.795113 250.643333 253.491553 256.339772
#[92] 259.187992 262.036212 264.884432 267.732651 270.580871 273.429091 276.277310
#[99] 279.125530 281.973750

#these are the times that BAMM has calculated our diversification rates at,
# so we have both diversification and extinction rates, not shown here,
#but in our BAMM object, that are calculated at each of these times for our phylogeny.
#these are the times labelled from 1-100. We've got 100 because BAMM automati-
#cally partitions the rate data into 100 equally spaced time bins.
#like all the other time series we've looked at, these are measured in units of millions
#of years. If you compare them to the data in your phylogeny, you should see that
#the oldest time here (time 100) = 281.973750 which is very close (nearly the same,
#just less decimal places) to the root of our tree (root.timeT = 281.9737).
#the 100th time is the oldest and the first time which is 0 is the present day.

#now let's take a look at lambda
rtt_T$lambda
#*lambda is how speciation is represented*
#we get a LOT of results
#let's just take a look at the length to see how many we have.
length(rtt_T$lambda)
#900100
#we expect this result because we had 9001 separate simulations of our
#diversification rates over time, and for each of those simulations
#we've got 100 times, so that gives us 900100.
```

[redacted]

```
#let's check the dimensions
dim(rtt_T$lambda)
#[1] 9001  100
#9001 times 100, which gives us the large figure of 900100.

#lets take a look at *extinction, which is represented by mu*
rtt_T$mu
#again, we have a huge number of results
#let's have a look at the dimensions to check that they're the same
dim(rtt_T$mu)
#[1] 9001  100
#and they are!

#the last thing we will look at is the "type"
rtt_T$type
#[1] "diversification"
#this refers to the type of analysis that was run to get these results.
#BAMM only has two types of analysis: diversification and trait.
#we're interested in evolutionary rates, therefore, these results are for the di-
#versificatoin analysis.

#let's move on to plotting and visualizing the rates through time

#PLOTTING TERRESTRIAL PSEUDOSUCHIA SPECIATION RATE TIME SERIES

#the following command is used to plot speciation rates (lambda)
terrestrial_speciation_rate_time_series_plot <- plotRateThroughTime(streeTerrestrial,
            ratetype='speciation',
            avgCol="blue",
            ylim=c(0,0.5),
            cex.axis=2,
            intervalCol='blue',
            intervals=c(0.05, 0.95),
            opacity=0.1)

#looking at it
terrestrial_speciation_rate_time_series_plot

#what we're doing here is telling R to plot speciation rates for our terrestrial
#subtree, and the other options are how we want it to be displayed
#in the plot is speciation rate on the y axis, measured in numbers of species
#per million years;
#on the x axis we have time, measured in millions of years.
```

[redacted]

```
#the solid line is our speciation rate through time and the lighter "shade""
#are the confidence intervals that we set in intervals.


#Now, let's save everything we've done
save.image("Crocs_Workshop3.RData")
#done

#Now or for next time we can start working on the marine taxa,
#and once you have, start thinking about what differences
#you might be seeing between the two habitat partitions
#you could also start to look at extinciton rates either for the terrstrial
#or for both the terrestrial and marine taxa
#you could also combine your outputs so far to see whether speciation rates appear to align with
major
#changes in the environmental data or with changes in the phylogeny


#PLOTTING TERRESTRIAL PSEUDOSUCHIA EXTINCTION RATE TIME SERIES

terrestrial_extinction_rate_time_series_plot <- plotRateThroughTime(streeTerrestrial,
                                          ratetype='extinction',
                                          avgCol="red",
                                          ylim=c(0,0.5),
                                          cex.axis=2,
                                          intervalCol='red',
                                          intervals=c(0.05, 0.95),
                                          opacity=0.1)

terrestrial_extinction_rate_time_series_plot

#
#
#
####Now doing the same but for the *Marine* taxa####
#
#
#
#

#*(speciation and extinction rates could not be adjusted the ~6 million years due to an issue
#with BAMMtools, so times for these plots were left as they were originally calculated*

#now, we can extract the rates for just our marine subtree
```

[redacted]

```
#first we need to tell BAMM to extract our subtree before it can get the rates.
streeMarine <- subtreeBAMM(edata, tips=MarineTaxa)
#here, we're saying that we want to extract the edata associated with just the marine taxa

#let's take a look at our extracted subtree
streeMarine
#
#Phylogenetic tree with 108 tips and 107 internal nodes.
#
#Tip labels:
# Atlantosuchus_coupatezi, Guarinisuchus_munizi, Rhabdognathus_acutirostris,
Rhabdognathus_aslerensis, Dyrosaurus_maghribensis, Dyrosaurus_phosphaticus, ...
#
#Rooted; includes branch lengths.
#
#Posterior samples: 9001
#
#List elements:
#  edge Nnode tip.label edge.length begin end downseq lastvisit numberEvents eventData
eventVectors tipStates tipLambda meanTipLambda eventBranchSegs tipMu meanTipMu type
#lets compare that to our subtree that we called treeM in our last workshop
treeM
#Phylogenetic tree with 108 tips and 107 internal nodes.
#
#Tip labels:
# Atlantosuchus_coupatezi, Guarinisuchus_munizi, Rhabdognathus_acutirostris,
Rhabdognathus_aslerensis, Dyrosaurus_maghribensis, Dyrosaurus_phosphaticus, ...
#
#Rooted; includes branch lengths.

#some of the data are the same, both of them have 108 tips and 107 internal nodes,
#and they contain the same taxa...
#but we now have some additional information in the BAMM subtree that we've just
#created: this also has our sample size ("Posterior samples: 9001", which is 9001
#because we have removed 10% as burn-in), and we also have a list of elements that
#our object contains. We won't be using all of these in this workshop, but you
#might want to look them up in the documentation to see what all of these do
#(there's a link to the documentation at the end of the workshop)

#now, let's extract the rates through time for our subtree
rtt_M <- getRateThroughTimeMatrix(streeMarine)

#now, let's take a look at what's inside our rates through time
summary(rtt_M)
```

[redacted]

```
#      Length Class  Mode
#lambda 900100 -none- numeric
#mu     900100 -none- numeric
#times     100 -none- numeric
#type       1 -none- character

#this shows us we have 4 elements: lambda, mu, times, type... let's go through each
#one
#first the times
rtt_M$times
#  [1]   0.000000   2.543582   5.087165   7.630747  10.174330  12.717912  15.261495
#[8]  17.805077  20.348660  22.892242  25.435825  27.979407  30.522989  33.066572
#[15]  35.610154  38.153737  40.697319  43.240902  45.784484  48.328067  50.871649
#[22]  53.415231  55.958814  58.502396  61.045979  63.589561  66.133144  68.676726
#[29]  71.220309  73.763891  76.307474  78.851056  81.394638  83.938221  86.481803
#[36]  89.025386  91.568968  94.112551  96.656133  99.199716 101.743298 104.286881
#[43] 106.830463 109.374045 111.917628 114.461210 117.004793 119.548375 122.091958
#[50] 124.635540 127.179123 129.722705 132.266287 134.809870 137.353452 139.897035
#[57] 142.440617 144.984200 147.527782 150.071365 152.614947 155.158530 157.702112
#[64] 160.245694 162.789277 165.332859 167.876442 170.420024 172.963607 175.507189
#[71] 178.050772 180.594354 183.137937 185.681519 188.225101 190.768684 193.312266
#[78] 195.855849 198.399431 200.943014 203.486596 206.030179 208.573761 211.117343
#[85] 213.660926 216.204508 218.748091 221.291673 223.835256 226.378838 228.922421
#[92] 231.466003 234.009586 236.553168 239.096750 241.640333 244.183915 246.727498
#[99] 249.271080 251.814663

#these are the times that BAMM has calculated our diversification rates at,
# so we have both diversification and extinction rates, not shown here,
#but in our BAMM object, that are calculated at each of these times for our phylogeny.
#these are the times labelled from 1-100.

#now let's take a look at lambda
rtt_M$lambda
#*lambda is how speciation is represented*
#we get a LOT of results
#let's just take a look at the length to see how many we have.
length(rtt_M$lambda)
#900100
#we expect this result because we had 9001 separate simulations of our
#diversification rates over time, and for each of those simulations
#we've got 100 times, so that gives us 900100.

#let's check the dimensions
dim(rtt_M$lambda)
```

[redacted]

```
#[1] 9001  100
#9001 times 100, which gives us the large figure of 900100.

#lets take a look at *extinction, which is represented by mu*
rtt_M$mu
#again, we have a huge number of results
#let's have a look at the dimensions to check that they're the same
dim(rtt_M$mu)
#[1] 9001  100
#and they are!

#the last thing we will look at is the "type"
rtt_M$type
#[1] "diversification"
#this refers to the type of analysis that was run to get these results.
#BAMM only has two types of analysis: diversification and trait.
#we're interested in evolutionary rates, therefore, these results are for the di-
#versificatoin analysis.

#PLOTTING MARINE PSEUDOSUCHIA SPECIATION RATE TIME SERIES
#the following command is used to plot speciation rates (lambda),
marine_speciation_rate_time_series_plot <- plotRateThroughTime(streeMarine,
                                           ratetype='speciation',
                                           avgCol="green",
                                           ylim=c(0,0.5),
                                           cex.axis=2,
                                           intervalCol='blue',
                                           intervals=c(0.05, 0.95),
                                           opacity=0.1)
#looking at it
marine_speciation_rate_time_series_plot

#what we're doing here is telling R to plot speciation rates for our marine
#subtree

#Now, let's save everything we've done
save.image("Crocs_Workshop3.RData")
#done

#PLOTTING MARINE PSEUDOSUCHIA EXTINCTION RATE TIME SERIES
marine_extinction_rate_time_series_plot <- plotRateThroughTime(streeMarine,
                                           ratetype='extinction',
                                           avgCol="pink",
                                           ylim=c(0,0.5),
```

[redacted]

```
                                                  cex.axis=2,
                                                  intervalCol='pink',
                                                  intervals=c(0.05, 0.95),
                                                  opacity=0.1)


marine_extinction_rate_time_series_plot
#troubleshooting
#there are some missing values in the extinction rates for
#marine taxa
#checking this is the issue
which(is.na(rtt_M$mu))
#it is this problem! There are lots of times for the marine
#taxa where the extinction rate values are NA.
rtt_M$mu[is.na(rtt_M$mu)] <- 0
#replotting  (the important part is to replace
#subtree with your variable name, which is rtt_M:
marine_extinction_rate_time_series_plot<-plotRateThroughTime(rtt_M,
            ratetype='extinction',
            avgCol="purple",
            ylim=c(0,0.5),
            cex.axis=2,
            intervalCol='blue',
            intervals=c(0.05, 0.95),
            opacity=0.1)


#Now, let's save everything we've done
save.image("Crocs_Workshop3.RData")

####################################################################################
#          6.                Statistical analyses to test whether                  #
#             environmental change drove diversification in Pseudosuchia          #
####################################################################################

#Loading data saved from previous workshops
#(only from workshop 1 and workshop 3 because
#workshop 3 data contains the data for both workshop 3 and 2)
load("Crocs_Workshop1.RData")
load("Crocs_Workshop3.RData")

#loading libraries
library(ggplot2)
```

[redacted]

```
#Loading the function that will be used to carry out
#the correlation analyses

DCCA <- function(x,y,s){
  xx<-cumsum(x)
  yy<-cumsum(y)
  t<-1:length(xx)
  F2sj_xy<-runif(floor(length(xx)/s))
  F2sj_xx<-F2sj_xy
  F2sj_yy<-F2sj_xy
  for(ss in seq(1,(floor(length(xx)/s)*s),by=s)){
    F2sj_xy[(ss-1)/s+1]<-sum((summary(lm(xx[ss:(ss+s-1)]~t[ss:(ss+s-
1)]))$residuals)*(summary(lm(yy[ss:(ss+s-1)]~t[ss:(ss+s-1)]))$residuals))/(s-1)
    F2sj_xx[(ss-1)/s+1]<-sum((summary(lm(xx[ss:(ss+s-1)]~t[ss:(ss+s-
1)]))$residuals)*(summary(lm(xx[ss:(ss+s-1)]~t[ss:(ss+s-1)]))$residuals))/(s-1)
    F2sj_yy[(ss-1)/s+1]<-sum((summary(lm(yy[ss:(ss+s-1)]~t[ss:(ss+s-
1)]))$residuals)*(summary(lm(yy[ss:(ss+s-1)]~t[ss:(ss+s-1)]))$residuals))/(s-1)
  }
  rho<-mean(F2sj_xy)/sqrt(mean(F2sj_xx)*mean(F2sj_yy))
  return(c(rho,1/sqrt(length(xx)),1-pnorm(abs(rho),mean=0,sd=1/sqrt(length(xx)))))
}

#An explanation of the DCCA function:
#
#DCCA (de-trended cross-correlation analysis)
#calculates a correlation coefficient between two time series
#by having a de-trended analysis,
#This is useful because some types of trends, longer term trends (e.g.
#cooling trend in temperature over the last 66 million years),
#can distort the data and lead to false correlations.
#And they need to be removed, as shorter-term fluctuations are the ones
#of interest. De-trending won't stop analyses from revealing correlations
#that are real, it will prevent those longer term trends from
#dominating so more confidence can be had on results.
#

#
#
#
#   Terrestrial Taxa Speciation rate and Temperature Correlation
#
#
#
```

[redacted]

```
#Because BAMM starts counting at zero and then
#goes backwards we need to flip the times in rtt_T

#To check the speciation rate time and rtt_T$times series actually don't align:
#
#Looking at the times from the BAMM object rtt_T
rtt_T$times
#the times start at 0 and end at 282mya
#
#looking at speciation rates for rtt_T by using colmeans(rtt_T$lambda)
#which gives us a mean value for each of the 9001 simulations
colMeans(rtt_T$lambda)
#
#by using the terrestrial speciation rate figure
#from the last workshop,for time 0, speciation rate= ~0.05 and for
#~282mya= slightly more than 0.3.
#
#the time and speciation rate values ARE in opposite orders.


#Flipping the times so that a correlation between
#the correct times and speciation rates is obtained
timesT = abs(rtt_T$times-max(rtt_T$times))
#(command= "substract the maximum length from each time
#and then take the absolute value")
#
#checking
timesT
# [1] 281.973750 279.125530 276.277310 273.429091 270.580871...
# ...[96]  11.392879   8.544659   5.696439   2.848220   0.000000
#
#times now start at 282 and finish at 0, so they have been
#successfully flipped


#To calculate how many simulations are
#in the diversification data (should equal 9001, the length
#of lambda divided by the number of times)
numberOfSims = length(rtt_T$lambda)/length(rtt_T$times)
#checking
numberOfSims
#[1] 9001
#perfect
```

[redacted]

```
#Correlations
#two sets of environmental data: global temperature and
#global sea level- we will run correlations for both.
#9001 simulations rather than just one time series.
#Ideally, all of these would be used for the correlations, but
#since that makes the analysis take a lot of time,
#the number of samples used for the correlation will be set to 100
#
#setting the sample size to 100
numberOfSamples = 100

#Setting up empty arrays to store correlation coefficients
#for the 4 terrestrial pseudosuchia analyses
#(for the speciation-temperature correlations)
cors_sptemp_T <- rep(NA, numberOfSamples)
#(for the speciation-sea level correlations)
cors_spseaLevel_T <- rep(NA, numberOfSamples)
#(for the extinction-temperature correlations)
cors_extemp_T <- rep(NA, numberOfSamples)
#(for the extinction-sea level correlations)
cors_exseaLevel_T <- rep(NA, numberOfSamples)
#
#(done using a rep function which returns an empty array filled
#with the value of x (NA) by y number of times (100 times, the sample size)
#array filled with NA and not 0 as this allows detection of errors
#(if code contains NA it fails, and what went wrong can
#be investigated, whereas if filled with 0s the code would
#work and problems can go unnoticed, affecting the results).

#Setting a random starting seed
#(to guarantee that all get the same results
#as each other and the same results as displayed in the workshop page
#every time that we run these 100 samples)
set.seed(1)

#Generating the random sample of 100
#drawn from the total 9001
samples = sample(1:numberOfSims, numberOfSamples, replace = FALSE)
#(code: "take the array (of 1 to 9001) and pull out a random
#number of samples (set to 100)" replace=FALSE stops it
#from putting in the same number twice
#
#Taking a look at which samples have been pulled out:
samples
```

[redacted]

```
#perfect

#Setting the count to 1
#(because only a sample of 100 will and not the full set,
#will be correlated,and we need to keep track of how many we've done)
count = 1

#Carrying out the correlation
#(using a for loop)

# This loops from 1 to the number of simulations, which is 9,001.
for (i in 1:numberOfSims ) {

  # Is it one of our samples(the 100 that we set previously)?
  if (i %in% samples){

    # If yes, do the correlation.
    # Start by interpolating the data. We do this because the two time series are different lengths.
    # We need them to start and end at the same times and we need the points in between to match up in order to carry out the correlation.

    # This line takes our speciation rates (lambda) and the corresponding times and interpolates the lambda onto the temperature times.
    #We can only do a correlation for the temperature data that we have so if we have more lambda times than temperature we cannot use them
    interpdivTsptemp = approx(timesT, rtt_T$lambda[i,], temperature$V1, method='linear', rule=1)
    #approx is the interpolation function and we set it to do a linear
    #interpolation; rule=1 is an option in the interpolation function and it
    #puts NAs where we don't have data- so we give it our speciation data
    #and corresponding times, and times where we want speciation data and
    #the temperature data temperature$V1, which are temperature times
    #interpdivTsptemp will now contain 2 elements: x and y, x is the temperature
    #times and y is our speciation rates at those times

    # Here we check whether there is a lambda for every temperature time, if not it's left as NA
    #we store this in a variable called end because we want to end it if we find any NAs
    end = which(is.na(interpdivTsptemp$y))


    # If we have no NAs, ie. there is a time in the temperature time series for each lambda,
    #we just use the interpolation as already calculated, and put them into
    #a new variable that we call div_rates; ft is our final temperature that
    #we set back in workshop 1.
    if (length(end) == 0) {
      div_rates = interpdivTsptemp$y
      ft = finaltemp

      # Otherwise, we only grab and use the times that have both lambda and temperature data
    } else {
      div_rates = interpdivTsptemp$y[-end]
      ft = finaltemp[-end]
    }

    # Now do the correlation using the interpolated data
    c = DCCA(as.numeric(unlist(div_rates)),as.numeric(unlist(ft)),
             length(ft)/10)
    #here we give it our two variables: div_rates and ft and we're making
    #sure that they are numeric arrays using as.numeric(unlist()); the
    #length(ft)/10) is where we tell it what window size to use and this
    #is just how we split up the data to do the correlation (this is a
    #standard size window that works for most data sets); DCCA is the
    #function that we set up at the start that carries out the correlation.


    # once done, we Store the correlation co-efficient in the empty array that
    #we set up earlier
    cors_sptemp_T[count] = c[1]

    # Increase your count by 1 ready for the next correlation
    count = count+1
  }
}


#Moving on to plotting results
#
#Using the ggplot2 library
#to plot the results as a histogram
terrestrial_sp_temp_correlation1=qplot(cors_sptemp_T,
                              geom="histogram",
                              bins=30,
                              fill=I("blue"))
#Having a look at the histogram
terrestrial_sp_temp_correlation1
#looks perfect
#
#making it look nicer
```

[redacted]

```
terrestrial_sp_temp_correlation2 = terrestrial_sp_temp_correlation1 +
  theme(panel.grid.major = element_blank(),
                  panel.grid.minor = element_blank(),
                  panel.background = element_blank(),
                  axis.line = element_line(colour = "black"),
       text = element_text(size=19))+
                  labs( x="Correlation Coefficient between Speciation
  Rate and Temperature through time",
                  y = "Count",
                  title =           "Speciation Rate and Temperature Correlation
            (Terrestrial Pseudosuchia)")
terrestrial_sp_temp_correlation2
#perfect


#Saving it to pdf using ggsave (also part of ggplot2)
ggsave(terrestrial_sp_temp_correlation2,file="SpeciationTerrestrialTemperature.pdf")

#Seeing if the relationship found is statistically significant
#
#By computing some summary stats on the correlation
#coefficients and then carrying out a test of
#significance, using sink to write all the statistics
#to one file.

#setting up sink, asking it to write to a file named
#"SpeciationTerrestrialTemperatureStats.txt"
sink(file="SpeciationTerrestrialTemperatureStats.txt")

# Calculate and print to file some summary statistics, e.g., median
#for that file
print(summary(cors_sptemp_T))

# Calculate and print the 95% confidence intervals
print(quantile(cors_sptemp_T, c(0.025, 0.975)))

# Carrying out a wilcoxon unpaired test to test for significance.
#This will calculate the wilcoxon test statistic and the p-value.
#(mu here is not the same as the mu mentioned elsewhere (extinction)
#Here, we are testing whether the distribution is symmetrical around 0.0 or not.
statsTsptemp <- (wilcox.test(cors_sptemp_T, mu=0.0, paired = FALSE))

#To print the p-value
statsTsptemp$p.value
```

```
#printing standard deviation
print(sd(cors_sptemp_T))

#Closing the sink
sink()


#when we look at the summary statistics saved in the text file we
#can note the first two lines as initial summary statistics;
#we also have our 95% confidence intervals and then we have our
#p-value at the bottom.


#
#
#
#
#Terrestrial Taxa Speciation rate and Sea Level Correlation
#
#
#
#

#restarting the counter for our next correlation
count = 1

# This loops from 1 to the number of simulations, which is 9,001.
for (i in 1:numberOfSims ) {

  # Is it one of our samples?
  if (i %in% samples){

    # If yes, do the correlation.
    # Again, we have to start by interpolating the data.
    interpdivTspsealevel = approx(timesT, rtt_T$lambda[i,], seaLevel$Age, method='linear', rule=1)
    end = which(is.na(interpdivTspsealevel$y))
    if (length(end) == 0) {
      div_rates = interpdivTspsealevel$y
      ft = seaLevel$SL
    } else {
      div_rates = interpdivTspsealevel$y[-end]
      ft = seaLevel$SL[-end]
    }

    # Now do the correlation
    c = DCCA(as.numeric(unlist(div_rates)),as.numeric(unlist(ft)),length(ft)/10)

    ## Store the correlation co-efficient
    cors_spseaLevel_T[count] = c[1]

    # Increase your count by 1 ready to do the next.
    count = count+1
  }
}

#plotting the results as a histogram
terrestrial_sp_sealevel_correlation1 = qplot(cors_spseaLevel_T,
                                          geom="histogram",
                                          bins=30,
                                          fill = I("blue"))
terrestrial_sp_sealevel_correlation1

#making it nicer
terrestrial_sp_sealevel_correlation2 = terrestrial_sp_sealevel_correlation1 +
  theme(panel.grid.major = element_blank(),
                    panel.grid.minor = element_blank(),
                    panel.background = element_blank(),
                    axis.line = element_line(colour = "black"),
        text = element_text(size=19))+
  labs( x="Correlation Coefficient between Speciation
  Rate and Sea Level through time",
        y = "Count",
        title =           "Speciation Rate and Sea Level Correlation
             (Terrestrial Pseudosuchia)")

#taking a look at it
terrestrial_sp_sealevel_correlation2
#looks perfect

#saving the plot to file
ggsave(terrestrial_sp_sealevel_correlation2,
      file="SpeciationTerrestrialSeaLevel.pdf")

#Calculating and saving some statistics to check our correlation
#is significant
sink(file="SpeciationTerrestrialSeaLevelStats.txt")
```

[redacted]

```
# Printing some summary stats
print(summary(cors_spseaLevel_T))

# Printing the 95% confidence intervals
print(quantile(cors_spseaLevel_T, c(0.025, 0.975)))

#Extracting the p-value
statsTspsealevel <- (wilcox.test(cors_spseaLevel_T, mu=0.0, paired = FALSE))

#Getting the p-value onto the file
statsTspsealevel$p.value

#Printing standard deviation
print(sd(cors_spseaLevel_T))

#Closing sink
sink()



#
#
#
#Terrestrial Taxa Extinction rate and Temperature Correlation
#
#
#
#

#DCCA function (to carry out analyses) already loaded
#timesT already set properly (flipped)

#To calculate how many simulations are
#in the diversification data (should equal 9001, the length
#of mu divided by the number of times)
numberOfSims = length(rtt_T$mu)/length(rtt_T$times)
#checking
numberOfSims
#[1] 9001
#perfect

#number of samples already set
#empty array for terrestrial extinction rate and temperature
#correlations already set, but checking:
```

[redacted]

```
cors_extemp_T

#setting a random starting seed- guaranteeing
#getting the same results
#every time that we run these 100 samples
set.seed(1)

#generating the random sample of 100 drawn from the total 9001 samples
samples = sample(1:numberOfSims, numberOfSamples, replace = FALSE)
#(code= "take the array (of 1 to 9001) and pull out a random
#number of samples (set to 100)", replace=FALSE stops it
#from putting in the same number twice)
#taking a look to see which samples it has pulled out:
samples

#Setting the count to 1
#(because only a sample of 100 will and not the full set,
#will be correlated,and we need to keep track of how many we've done)
count = 1

#Correlation using a for loop (which loops from 1 to
#the number of simulations, which is 9,001)
for (i in 1:numberOfSims ) {

  # Is it one of our samples(the 100 that we set previously)?
  if (i %in% samples){

    # If yes, do the correlation.
    # Start by interpolating the data. We do this because the two time series are different lengths.
    # We need them to start and end at the same times and we need the points in between to
match up in order to carry out the correlation.

    # This line takes our extinction rates (mu) and the corresponding times and interpolates the
mu onto the temperature times.
    #We can only do a correlation for the temperature data that we have so if we have more mu
times than temperature we cannot use them
    interpdivTextemp = approx(timesT, rtt_T$mu[i,], temperature$V1, method='linear', rule=1)
    #approx is the interpolation function and we set it to do a linear
    #interpolation; rule=1 is an option in the interpolation function and it
    #puts NAs where we don't have data- so we give it our extinction data
    #and corresponding times, and times where we want extinction data and
    #the temperature data temperature$V1, which are temperature times.
    #interpdivTextemp will now contain 2 elements: x and y, x is the temperature
    #times and y is our extinction rates at those times
```

[redacted]

```
    # Here we check whether there is a mu for every temperature time, if not it's left as NA
    #we store this in a variable called end because we want to end it if we find any NAs
    end = which(is.na(interpdivTextemp$y))


    # If we have no NAs, ie. there is a time in the temperature time series for each mu,
    #we just use the interpolation as already calculated, and put them into
    #a new variable that we call div_rates; ft is our final temperature that
    #we set back in workshop 1.
    if (length(end) == 0) {
      div_rates = interpdivTextemp$y
      ft = finaltemp

      # Otherwise, we only grab and use the times that have both mu and temperature data
    } else {
      div_rates = interpdivTextemp$y[-end]
      ft = finaltemp[-end]
    }

    # Now do the correlation using the interpolated data
    c = DCCA(as.numeric(unlist(div_rates)),as.numeric(unlist(ft)),
             length(ft)/10)
    #here we give it our two variables: div_rates and ft and we're making
    #sure that they are numeric arrays using as.numeric(unlist()); the
    #length(ft)/10) is where we tell it what window size to use and this
    #is just how we split up the data to do the correlation (this is a
    #standard size window that works for most data sets); DCCA is the
    #function that we set up at the start that carries out the correlation.


    # once done, we Store the correlation co-efficient in the empty array that
    #we set up earlier
    cors_extemp_T[count] = c[1]

    # Increase your count by 1 ready for the next correlation
    count = count+1
  }
}

#Terrestrial-extinction-temperature
#when we run the code above we get an error message
warnings()
#1: In summary.lm(lm(xx[ss:(ss + s - 1)] ~ t[ss:(ss + s -  ... :
```

[redacted]

```
#essentially perfect fit: summary may be unreliable
#we can ignore it


#Moving on to plotting our results
#Using the ggplot2 library to plot the results as a histogram
terrestrial_extinction_temperature_correlation_plot1=qplot(cors_extemp_T,
                                                 geom="histogram",
                                                 bins=30,
                                                 fill= I("#990033"))

terrestrial_extinction_temperature_correlation_plot1
#looks good

#making it look nicer
terrestrial_extinction_temperature_correlation_plot2                                  =
terrestrial_extinction_temperature_correlation_plot1 +
  theme(panel.grid.major = element_blank(),
                  panel.grid.minor = element_blank(),
                  panel.background = element_blank(),
                  axis.line = element_line(colour = "black"),
       text = element_text(size=19))+
  labs( x="Correlation Coefficient between Extinction
  Rate and Temperature through time",
       y = "Count",
       title =          "Extinction Rate and Temperature Correlation
            (Terrestrial Pseudosuchia)")
terrestrial_extinction_temperature_correlation_plot2
#looks much better

#Saving it to pdf using ggsave (also part of ggplot2)
ggsave(terrestrial_extinction_temperature_correlation_plot2,
      file="ExtinctionTerrestrialTemperature.pdf")

#Seeing if the relationship found is statistically significant
#(By computing some summary stats on the correlation
#coefficients and then carrying out a test of
#significance, using sink to write all the statistics
#to one file)
#
#setting up sink, asking it to write to a file named
#"SpeciationTerrestrialTemperatureStats.txt"
sink(file="ExtinctionTerrestrialTemperatureStats.txt")
```

[redacted]

```
# Calculate and print to file some summary statistics, e.g., median
#for that file
print(summary(cors_extemp_T))

# Calculate and print the 95% confidence intervals
print(quantile(cors_extemp_T, c(0.025, 0.975)))

# Carrying out a wilcoxon unpaired test to test for significance.
#This will calculate the wilcoxon test statistic and the p-value.
#Beware! The mu here is not the same as the mu mentioned elsewhere (extinction).
#Here, we are testing whether the distribution is symmetrical around 0.0 or not.
statsTextemp <- (wilcox.test(cors_extemp_T, mu=0.0, paired = FALSE))

# We want to know the p-value
statsTextemp$p.value

#Printing standard deviation
print(sd(cors_extemp_T))

#Closing the sink
sink()


#
#
#
#
#Terrestrial Taxa Extinction rate and Sea Level Correlation
#
#
#
#

#restarting the counter for our next correlation
count = 1

# This loops from 1 to the number of simulations, which is 9,001.
for (i in 1:numberOfSims ) {

  # Is it one of our samples?
  if (i %in% samples){

    # If yes, do the correlation.
    # Again, we have to start by interpolating the data.
    interpdivTexsealevel = approx(timesT, rtt_T$mu[i,], seaLevel$Age, method='linear', rule=1)
    end = which(is.na(interpdivTexsealevel$y))
    if (length(end) == 0) {
      div_rates = interpdivTexsealevel$y
      ft = seaLevel$SL
    } else {
      div_rates = interpdivTexsealevel$y[-end]
      ft = seaLevel$SL[-end]
    }

    # Now do the correlation
    c = DCCA(as.numeric(unlist(div_rates)),as.numeric(unlist(ft)),length(ft)/10)

    ## Store the correlation co-efficient
    cors_exseaLevel_T[count] = c[1]

    # Increase your count by 1 ready to do the next.
    count = count+1
  }
}

#Terrestrial-extinction-sea level
#after running the command above we get this warning message which refers to the DCCA
function
warnings()
#1: In summary.lm(lm(xx[ss:(ss + s - 1)] ~ t[ss:(ss + s -  ... :
#essentially perfect fit: summary may be unreliable...
#we can ignore this

#plotting the results as a histogram
terrestrial_extinction_rate_sea_level_correlation1 =
  qplot(cors_exseaLevel_T,
        geom="histogram",
        bins=30,
        fill= I("#990033"))
#taking a look at it
terrestrial_extinction_rate_sea_level_correlation1

#making it nicer
terrestrial_extinction_rate_sea_level_correlation2 =
terrestrial_extinction_rate_sea_level_correlation1 + theme(panel.grid.major = element_blank(),
                                               panel.grid.minor = element_blank(),
                                               panel.background = element_blank(),
                                               axis.line = element_line(colour = "black"),
                                    text = element_text(size=19))+
  labs( x="Correlation Coefficient between Extinction
  Rate and Sea Level through time",
       y = "Count",
       title =           "Extinction Rate and Sea Level Correlation
            (Terrestrial Pseudosuchia)")

#taking a look at it
terrestrial_extinction_rate_sea_level_correlation2

#saving the plot to file
ggsave(terrestrial_extinction_rate_sea_level_correlation2,
      file="ExtinctionTerrestrialSeaLevel.pdf")

#Calculating and saving some statistics to check our correlation
#is significant
sink(file="ExtinctionTerrestrialSeaLevelStats.txt")

# Printing some summary stats
print(summary(cors_exseaLevel_T))

# Printing the 95% confidence intervals
print(quantile(cors_exseaLevel_T, c(0.025, 0.975)))

#Extracting the p-value
statsexseaLevel_T <- (wilcox.test(cors_exseaLevel_T, mu=0.0, paired = FALSE))

#Getting the p-value onto the file
statsexseaLevel_T$p.value

#Printing the standard deviation
print(sd(cors_exseaLevel_T))

#Closing the sink
sink()



#
#
#
#
#
#
```

[redacted]

```
#
##################   Now for the MARINE TAXA  #####################
#
#
#
#
#
#
#
#

#
#
#
#Marine Taxa Speciation rate and Temperature Correlation
#
#
#

#Before doing anything with the marine taxa, we need
#to fix the times associated with the speciation/extinction rates,
#by adding 6.2895 MY to each of them. (previously discussed
#issue with the times= marine taxa records should not
#reach the present time)

rtt_M$times <- rtt_M$times + 6.2895

#Because BAMM starts counting at 6.2895 and then
#goes backwards we need to flip the times in rtt_M

#To check the speciation rate time and rtt_M$times time series
#actually don't align:

#Looking at the times from the BAMM object rtt_M
rtt_M$times
#the times start at  6.2895mya and end at ~258mya
#[1]   6.289500   8.833082  11.376665  13.920247  16.463830  19.007412  21.550995  24.094577
26.638160
# [97] 250.473415 253.016998 255.560580 258.104163

#looking at speciation rates for rtt_M by using colmeans(rtt_M$lambda)
#which gives us a mean value for each of the 9001 simulations
colMeans(rtt_M$lambda)
#
```

[redacted]

```
#by using the marine speciation rate figure
#produced earlier, for time closest to present,
#speciation rate= ~0.05 and for
#oldest time= ~0.15
#
#the time and speciation rate values ARE in opposite orders.

#Flipping the times so that a correlation between
#the correct times and speciation rates is obtained
timesM = abs(rtt_M$times-max(rtt_M$times+6.289500))
#(command= "substract the maximum length from each time
#and then take the absolute value")
#
#checking
timesM
# [1] 258.104163 255.560580 253.016998 250.473415 ...
#...[97]  13.920247  11.376665   8.833082   6.289500
#they now start at 251.8 and finish at 0, so the times have been
#successfully flipped


#Calculating how many simulations are in
#our diversification data (should equal 9001, the length
#of lambda divided by the number of times)
numberOfSims = length(rtt_M$lambda)/length(rtt_M$times)
#checking
numberOfSims
#[1] 9001
#perfect

#Correlations
#two sets of environmental data: global temperature and
#global sea level- we will run correlations for both.
#9001 simulations rather than just one time series.
#Ideally, all of these would be used for the correlations, but
#since that makes the analysis take a lot of time,
#the number of samples used for the correlation will be set to 100
#
#setting the sample size to 100
numberOfSamples = 100

##Setting up empty arrays to store correlation coefficients
#for the 4 marine pseudosuchia analyses
#(for the speciation-temperature correlations)
```

[redacted]

```
cors_sptemp_M <- rep(NA, numberOfSamples)
#(for the speciation-sea level correlations)
cors_spseaLevel_M <- rep(NA, numberOfSamples)
#(for the extinction-temperature correlations)
cors_extemp_M <- rep(NA, numberOfSamples)
#(for the extinction-sea level correlations)
cors_exseaLevel_M <- rep(NA, numberOfSamples)
#
#(done using a rep function which returns an empty array filled
#with the value of x (NA) by y number of times (100 times, the sample size).
#Array filled with NA and not 0 as this allows detection of errors
#(if code contains NA it fails, and what went wrong can
#be investigated, whereas if filled with 0s the code would
#work and problems can go unnoticed, affecting the results).

#Setting a random starting seed to guarantee that
#we get the same results
#every time that we run these 100 samples
set.seed(1)

#Generating our random sample of 100
#drawn from the 9001 that we have in total
samples = sample(1:numberOfSims, numberOfSamples, replace = FALSE)
#(code: "take our array (of 1 to 9001) and pull out a random
#number of samples (set to 100)", replace=FALSE stops it
#from putting in the same number twice
#
#Seeing which samples it has pulled out:
samples


#Setting the count to 1
#(because only a sample of 100 will and not the full set,
#will be correlated,and we need to keep track of how many we've done)
count = 1

#Carrying out the correlation
#(using a for loop)

# This loops from 1 to the number of simulations, which is 9,001.
for (i in 1:numberOfSims ) {

  # Is it one of our samples(the 100 that we set previously)?
  if (i %in% samples){
```

[redacted]

```
  # If yes, do the correlation.
  # Start by interpolating the data. We do this because the two time series are different lengths.
  # We need them to start and end at the same times and we need the points in between to
match up in order to carry out the correlation.

  # This line takes our speciation rates (lambda) and the corresponding times and interpolates
the lambda onto the temperature times.
  #We can only do a correlation for the temperature data that we have so if we have more
lambda times than temperature we cannot use them
  interpdivMsptemp = approx(timesM, rtt_M$lambda[i,], temperature$V1, method='linear',
rule=1)
  #approx is the interpolation function and we set it to do a linear
  #interpolation; rule=1 is an option in the interpolation function and it
  #puts NAs where we don't have data- so we give it our speciation data
  #and corresponding times, and times where we want speciation data and
  #the temperature data temperature$V1, which are temperature times
  #interpdivMsptemp will now contain 2 elements: x and y, x is the temperature
  #times and y is our speciation rates at those times

  # Here we check whether there is a lambda for every temperature time, if not it's left as NA
  #we store this in a variable called end because we want to end it if we find any NAs
  end = which(is.na(interpdivMsptemp$y))


  # If we have no NAs, ie. there is a time in the temperature time series for each lambda,
  #we just use the interpolation as already calculated, and put them into
  #a new variable that we call div_rates; ft is our final temperature that
  #we set back in workshop 1.
  if (length(end) == 0) {
    div_rates = interpdivMsptemp$y
    ft = finaltemp

    # Otherwise, we only grab and use the times that have both lambda and temperature data
  } else {
    div_rates = interpdivMsptemp$y[-end]
    ft = finaltemp[-end]
  }

  # Now do the correlation using the interpolated data
  c = DCCA(as.numeric(unlist(div_rates)),as.numeric(unlist(ft)),
          length(ft)/10)
  #here we give it our two variables: div_rates and ft and we're making
  #sure that they are numeric arrays using as.numeric(unlist()); the
```

[redacted]

```
    #length(ft)/10) is where we tell it what window size to use and this
    #is just how we split up the data to do the correlation (this is a
    #standard size window that works for most data sets); DCCA is the
    #function that we set up at the start that carries out the correlation.


    # once done, we Store the correlation co-efficient in the empty array that
    #we set up earlier
    cors_sptemp_M[count] = c[1]

    # Increase your count by 1 ready for the next correlation
    count = count+1
  }
}



#Plotting our results
#Using the ggplot2 library to plot the results as a histogram
marine_sp_temp_correlation1=qplot(cors_sptemp_M,
                                  geom="histogram",
                                  bins=30,
                                  fill=I("blue"))
#taking a look at the histogram
marine_sp_temp_correlation1
#looks good

#making it nicer
marine_sp_temp_correlation2 = marine_sp_temp_correlation1 +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_line(colour = "black"),
        text = element_text(size=19))+
  labs( x="Correlation Coefficient between Speciation
  Rate and Temperature through time",
        y = "Count",
        title =   "Speciation Rate and Temperature Correlation
               (Marine Pseudosuchia)")

#taking a look at it
marine_sp_temp_correlation2
#looks perfect

#Saving it to pdf using ggsave (also part of ggplot2)
```

[redacted]

```
ggsave(marine_sp_temp_correlation2,file="SpeciationMarineTemperature.pdf")

##Calculating and saving some statistics to check our correlation
#is significant
#
#setting up sink, asking it to write to a file named
#"SpeciationMarineTemperatureStats.txt"
sink(file="SpeciationMarineTemperatureStats.txt")

# Calculate and print to file some summary statistics, e.g., median
#for that file
print(summary(cors_sptemp_M))

# Calculate and print the 95% confidence intervals
print(quantile(cors_sptemp_M, c(0.025, 0.975)))

# Carrying out a wilcoxon unpaired test to test for significance.
#This will calculate the wilcoxon test statistic and the p-value.
#Beware! The mu here is not the same as the mu mentioned elsewhere (extinction).
#Here, we are testing whether the distribution is symmetrical around 0.0 or not.
statssptemp_M <- (wilcox.test(cors_sptemp_M, mu=0.0, paired = FALSE))

#Getting the p-value
statssptemp_M$p.value

#Printing out the standard deviation
print(sd(cors_sptemp_M))

#closing the sink
sink()




#
#
#
#
#
# Marine Taxa Speciation rate and Sea Level Correlation
#
#
#
#
```

[redacted]

```
#restarting the counter for our next correlation
count = 1

# This loops from 1 to the number of simulations, which is 9,001.
for (i in 1:numberOfSims ) {

  # Is it one of our samples?
  if (i %in% samples){

    # If yes, do the correlation.
    # Again, we have to start by interpolating the data.
    interpdivMspsealevel = approx(timesM, rtt_M$lambda[i,], seaLevel$Age, method='linear', rule=1)
    end = which(is.na(interpdivMspsealevel$y))
    if (length(end) == 0) {
      div_rates = interpdivMspsealevel$y
      ft = seaLevel$SL
    } else {
      div_rates = interpdivMspsealevel$y[-end]
      ft = seaLevel$SL[-end]
    }

    # Now do the correlation
    c = DCCA(as.numeric(unlist(div_rates)),as.numeric(unlist(ft)),length(ft)/10)

    ## Store the correlation co-efficient
    cors_spseaLevel_M[count] = c[1]

    # Increase your count by 1 ready to do the next.
    count = count+1
  }
}

#plotting the results as a histogram
marine_sp_sealevel_correlation1 = qplot(cors_spseaLevel_M,
                                         geom="histogram",
                                         bins=30,
                                         fill = I("blue"))
#taking a look at the histogram
marine_sp_sealevel_correlation1

#making it nicer
marine_sp_sealevel_correlation2 = marine_sp_sealevel_correlation1 +
  theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.background = element_blank(),
      axis.line = element_line(colour = "black"),
      text = element_text(size=19))+
  labs( x="Correlation Coefficient between Speciation
  Rate and Sea Level through time",
      y = "Count",
      title =         "Speciation Rate and Sea Level Correlation
          (Marine Pseudosuchia)")

#taking a look at it
marine_sp_sealevel_correlation2

#saving the plot to file
ggsave(marine_sp_sealevel_correlation2,
     file="SpeciationMarineSeaLevel.pdf")

#Calculating and saving some statistics to check our correlation
#is significant
#
#setting up sink, asking it to write to a file named
#"SpeciationMarineSeaLevelStats.txt"
sink(file="SpeciationMarineSeaLevelStats.txt")

# Printing some summary stats
print(summary(cors_spseaLevel_M))

# Printing the 95% confidence intervals
print(quantile(cors_spseaLevel_M, c(0.025, 0.975)))

#Extracting the p-value
statsspseaLevel_M <- (wilcox.test(cors_spseaLevel_M, mu=0.0, paired = FALSE))

#Getting the p-value onto the file
statsspseaLevel_M$p.value

#printing standard deviation
print(sd(cors_spseaLevel_M))

#Closing the sink
sink()


#
```

[redacted]

```
#
#
#Marine Taxa Extinction rate and Temperature Correlation
#
#
#

#DCCA function (to carry out analyses) already loaded
#timesM already set

#To calculate how many simulations are
#in the diversification data (should equal 9001, the length
#of mu divided by the number of times)
numberOfSims = length(rtt_M$mu)/length(rtt_M$times)
#checking
numberOfSims
#[1] 9001
#perfect

#number of samples already set
#
#empty array for marine extinction and temperature correlation
#already set, but checking:
cors_extemp_M

#setting a random starting seed- guaranteeing
#getting the same results
#every time that we run these 100 samples
set.seed(1)

#generating the random sample of 100
#drawn from the total 9001 samples
samples = sample(1:numberOfSims, numberOfSamples, replace = FALSE)
#this is saying to take our array (of 1 to 9001) and pull out a random
#number of samples that we've set to 100 and replace=FALSE stops it
#from putting in the same number twice
#we can take a look and see which samples it has pulled out like this:
samples


#Setting the count to 1
#(because only a sample of 100 will and not the full set,
#will be correlated,and we need to keep track of how many we've done)
count = 1
```

[redacted]

```
#Carrying out the correlation
#(using a for loop)

# This loops from 1 to the number of simulations, which is 9,001.
for (i in 1:numberOfSims ) {

  # Is it one of our samples(the 100 that we set previously)?
  if (i %in% samples){

    # If yes, do the correlation.
    # Start by interpolating the data. We do this because the two time series are different lengths.
    # We need them to start and end at the same times and we need the points in between to
match up in order to carry out the correlation.

    # This line takes our extinction rates (mu) and the corresponding times and interpolates the
mu onto the temperature times.
    #We can only do a correlation for the temperature data that we have so if we have more mu
times than temperature we cannot use them
    interpdivMextemp = approx(timesM, rtt_M$mu[i,], temperature$V1, method='linear', rule=1)
    #approx is the interpolation function and we set it to do a linear
    #interpolation; rule=1 is an option in the interpolation function and it
    #puts NAs where we don't have data- so we give it our extinction data
    #and corresponding times, and times where we want extinction data and
    #the temperature data temperature$V1, which are temperature times
    #interpdivMextemp will now contain 2 elements: x and y, x is the temperature
    #times and y is our extinction rates at those times

    # Here we check whether there is a mu for every temperature time, if not it's left as NA
    #we store this in a variable called end because we want to end it if we find any NAs
    end = which(is.na(interpdivMextemp$y))


    # If we have no NAs, ie. there is a time in the temperature time series for each mu,
    #we just use the interpolation as already calculated, and put them into
    #a new variable that we call div_rates; ft is our final temperature that
    #we set back in workshop 1.
    if (length(end) == 0) {
      div_rates = interpdivMextemp$y
      ft = finaltemp

      # Otherwise, we only grab and use the times that have both mu and temperature data
    } else {
      div_rates = interpdivMextemp$y[-end]
```

[redacted]

```
      ft = finaltemp[-end]
    }

    # Now do the correlation using the interpolated data
    c = DCCA(as.numeric(unlist(div_rates)),as.numeric(unlist(ft)),
             length(ft)/10)
    #here we give it our two variables: div_rates and ft and we're making
    #sure that they are numeric arrays using as.numeric(unlist()); the
    #length(ft)/10) is where we tell it what window size to use and this
    #is just how we split up the data to do the correlation (this is a
    #standard size window that works for most data sets); DCCA is the
    #function that we set up at the start that carries out the correlation.


    # once done, we Store the correlation co-efficient in the empty array that
    #we set up earlier
    cors_extemp_M[count] = c[1]

    # Increase your count by 1 ready for the next correlation
    count = count+1
  }
}

#Marine-extinction-temperature
#after running this command we get a warning message
warnings()
#1: In summary.lm(lm(xx[ss:(ss + s - 1)] ~ t[ss:(ss + s -  ... :
#essentially perfect fit: summary may be unreliable
#we can ignore this


#now we can move on to plotting our results, using the ggplot2
#library to plot the results as a histogram
marine_extinction_temperature_correlation_plot1=qplot(cors_extemp_M,
                                                      geom="histogram",
                                                      bins=30,
                                                      fill= I("#990033"))
#taking a look at it
marine_extinction_temperature_correlation_plot1

#making it look nicer
marine_extinction_temperature_correlation_plot2                                      =
marine_extinction_temperature_correlation_plot1 +
  theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.background = element_blank(),
      axis.line = element_line(colour = "black"),
      text = element_text(size=19))+
  labs( x="Correlation Coefficient between Extinction
  Rate and Temperature through time",
      y = "Count",
      title =          "Extinction Rate and Temperature Correlation
          (Marine Pseudosuchia)")
#taking a look at it
marine_extinction_temperature_correlation_plot2


#saving it to pdf using ggsave (also part of ggplot2)
ggsave(marine_extinction_temperature_correlation_plot2,
      file="ExtinctionMarineTemperature.pdf")

#Calculating and saving some statistics to check our correlation
#is significant
#setting up sink, asking it to write to a file named
#"SpeciationTerrestrialTemperatureStats.txt"
sink(file="ExtinctionMarineTemperatureStats.txt")

# Calculate and print to file some summary statistics, e.g., median
#for that file
print(summary(cors_extemp_M))

# Calculate and print the 95% confidence intervals
print(quantile(cors_extemp_M, c(0.025, 0.975)))

# Carrying out a wilcoxon unpaired test to test for significance.
#This will calculate the wilcoxon test statistic and the p-value.
#Beware! The mu here is not the same as the mu mentioned elsewhere (extinction).
#Here, we are testing whether the distribution is symmetrical around 0.0 or not.
statsextemp_M <- (wilcox.test(cors_extemp_M, mu=0.0, paired = FALSE))

# We want to know the p-value
statsextemp_M$p.value

#Printing the standard deviation
print(sd(cors_extemp_M))

#closing the sink function
sink()
```

[redacted]

```
#
#
#
#
#
#Marine Taxa Extinction rate and Sea Level Correlation
#
#
#
#

#restarting the counter for our next correlation
count = 1

# This loops from 1 to the number of simulations, which is 9,001.
for (i in 1:numberOfSims ) {

 # Is it one of our samples?
 if (i %in% samples){

  # If yes, do the correlation.
  # Again, we have to start by interpolating the data.
  interpdivMexsealevel = approx(timesM, rtt_M$mu[i,], seaLevel$Age, method='linear', rule=1)
  end = which(is.na(interpdivMexsealevel$y))
  if (length(end) == 0) {
   div_rates = interpdivMexsealevel$y
   ft = seaLevel$SL
  } else {
   div_rates = interpdivMexsealevel$y[-end]
   ft = seaLevel$SL[-end]
  }

  # Now do the correlation
  c = DCCA(as.numeric(unlist(div_rates)),as.numeric(unlist(ft)),length(ft)/10)

  ## Store the correlation co-efficient
  cors_exseaLevel_M[count] = c[1]

  # Increase your count by 1 ready to do the next.
  count = count+1
 }
}
```

[redacted]

```
#Marine-extinction-sea level
#we get a warning message
warnings()
#1: In summary.lm(lm(xx[ss:(ss + s - 1)] ~ t[ss:(ss + s -  ... :
#essentially perfect fit: summary may be unreliable
#we can ignore this

#plotting the results as a histogram
marine_extinction_rate_sea_level_correlation1 =
  qplot(cors_exseaLevel_M,
        geom="histogram",
        bins=30,
        fill= I("#990033"))
#taking a look at it
marine_extinction_rate_sea_level_correlation1

#making it nicer
marine_extinction_rate_sea_level_correlation2                                        =
marine_extinction_rate_sea_level_correlation1 + theme(panel.grid.major = element_blank(),
                                                       panel.grid.minor              =
element_blank(),
                                                       panel.background              =
element_blank(),
                                                       axis.line  =  element_line(colour  =
"black"),
                                                   text = element_text(size=19))+
  labs( x="Correlation Coefficient between Extinction
  Rate and Sea Level through time",
        y = "Count",
        title =          "Extinction Rate and Sea Level Correlation
           (Marine Pseudosuchia)")
#taking a look at it
marine_extinction_rate_sea_level_correlation2

#saving the plot to file
ggsave(marine_extinction_rate_sea_level_correlation2,
       file="ExtinctionMarineSeaLevel.pdf")

#Calculating and saving some statistics to check our correlation
#is significant
#setting up sink, asking it to write to a file named
#"ExtinctionMarineSeaLevelStats.txt"
sink(file="ExtinctionMarineSeaLevelStats.txt")
```

[redacted]

```
# Printing some summary stats
print(summary(cors_exseaLevel_M))

# Printing the 95% confidence intervals
print(quantile(cors_exseaLevel_M, c(0.025, 0.975)))

#Extracting the p-value
statsexseaLevel_M <- (wilcox.test(cors_exseaLevel_M, mu=0.0, paired = FALSE))

#Getting the p-value onto the file
statsexseaLevel_M$p.value

#printing standard deviation
print(sd(cors_exseaLevel_M))
#Closing the sink
sink()
#Saving the data we have so far
save.image("Crocs_Workshop4.RData")
```