---

title: \Huge \textbf{Species sympatry shapes brain size evolution in Primates}

# date: "`r format(Sys.time(), '%B %d, %Y')`"

#I follow: https://stackoverflow.com/questions/52918716/authors-and-affiliations-in-the-yaml-of-rmarkdown for author display

author:

- name: \textit{Benjamin Robira}

email: benjamin.robira@normalesup.org

institute: [CEFE, MH]

correspondence: true

- name: \textit{Benoit Perez-Lamarque}

email: benoit.perez@ens.psl.eu

institute: [ENS, MNHN]

institute:

- CEFE: Centre d’Écologie Fonctionnelle et Évolutive, Université de Montpellier & CNRS, Montpellier, France.

- MH: Anthropologie et Ethnobiologie, Centre National de la Recherche Scientifique/Muséum National d'Histoire Naturelle, University Paris Diderot, Sorbonne Paris Cité, Musée de l'Homme, Paris, France.

- ENS: Institut de Biologie de l’École Normale Supérieure (IBENS), École Normale Supérieure, CNRS, INSERM, Université PSL, Paris, France.

- MNHN: Institut de Systématique, Évolution, Biodiversité (ISYEB), Muséum National d’Histoire Naturelle, CNRS, Sorbonne Université, EPHE, Université des Antilles, Paris, France.

# output:

# # - '--lua-filter=scholarly-metadata.lua'

# # - '--lua-filter=author-info-blocks.lua'

# bookdown::pdf\_book:

# #bookdown::word\_document2:

# number\_sections: false

# toc: false

# #citation\_package: natbib

# latex\_engine: pdflatex

# fig\_caption: true

# pandoc\_args:

# - '--lua-filter=scholarly-metadata.lua'

# - '--lua-filter=author-info-blocks.lua'

#

# bibliography: bibliographyarticlepackage.bib

# csl: nature.csl

# #biblio-style: apa

# always\_allow\_html: true

# link-citations: yes

output:

# - '--lua-filter=scholarly-metadata.lua'

# - '--lua-filter=author-info-blocks.lua'

bookdown::pdf\_book:

#bookdown::word\_document2:

number\_sections: false

toc: false

citation\_package: biblatex

latex\_engine: pdflatex

fig\_caption: true

pandoc\_args:

- '--lua-filter=scholarly-metadata.lua'

- '--lua-filter=author-info-blocks.lua'

always\_allow\_html: true

bibliography: bibliographyarticlepackage.bib

biblio-style: nature

biblatexoptions: [natbib=true]

urlcolor: blue

filecolor: blue

linkcolor: blue

fontsize: 12pt

header-includes:

- \usepackage{lettrine}

# - \usepackage[nolists, nomarkers,tablesfirst]{endfloat} # For figures and tables at end

#- \usepackage{lineno} # For line numbering

#- \linenumbers # For line numbering

#- \usepackage{setspace}\doublespacing

- \usepackage{fontawesome} #for fa symbols

- \usepackage{tcolorbox}

- \pagenumbering{gobble} #for no page numbering

- \setlength{\parskip}{0em} #to remove line gaps

- \DeclareUnicodeCharacter{2212}{-}

- \usepackage{caption}

- \captionsetup[figure]{font=small}

- \newcommand{\beginsupplement}{

\setcounter{table}{0}

\renewcommand{\thetable}{S\arabic{table}}

\setcounter{figure}{0}

\renewcommand{\thefigure}{S\arabic{figure}}

}

---

<!-- ------------------------------------------------------------------- -->

<!-- --------------------- LAYOUT --------------------------- -->

<!-- ------------------------------------------------------------------- -->

<!-- TO PUT THE LETTRINE -->

\newcommand{\initial}[1]{%

\lettrine[lraise=0, loversize=0.5,nindent=0em]{

\color{black}

{\textsc{#1}}}{}}

<!-- TO CREATE GRAY BOX -->

\newtcolorbox{graybox}{

colback=lightgray!25,

colframe=lightgray!25,

coltext=black,

boxsep=3pt,

arc=0pt}

<!-- ------------------------------------------------------------------- -->

<!-- ------------------------------------------------------------------- -->

<!-- ------------------------------------------------------------------- -->

```{r setup, include=FALSE, echo=FALSE, message=FALSE}

knitr::opts\_chunk$set(include=FALSE, echo=FALSE, message=FALSE)

knitr::opts\_chunk$set(dpi=300) # Figure resolution and size out.width = '100%',

knitr::opts\_chunk$set(fig.pos = 'p'#, fig.align = 'center'

) # Places figures on pages separate from text, centered

knitr::opts\_chunk$set(fig.env="figure") # Latex figure environment

knitr::opts\_knit$set(eval.after = "fig.cap") #To insert R code into R figure caption

```

<!-- TC:ignore -->

```{r, echo=FALSE, results= 'hide'}

#Import librairies

library(readr)

#Plot

library(RColorBrewer)

library(tidyr)

library(stringr)

library(svMisc)

library(plotrix)

library(circlize)

#Spatial

library(rworldmap) # World map

library(cleangeo) #to clean it otherwise issues with intersection

library(maps)

library(rgeos) #for readOGR; gArea/gCentroid...

library(sf) #for intersection

library(rgdal)

library(geosphere)

#Phylogeny

library(phytools)

library(ape)

library(phylolm)

#Segmentation

library(strucchange)

#Import own function

source("T:/Saved\_PhD/Empirical\_analysis/Scripts&Functions/Functions/toolbox.R", local = knitr::knit\_global())

#Create citation fusion between articles and package (based on toolbox function)

citeR(

bibliographyArticle="T:/Saved\_PhD/Library\_general/libraryMdf.bib",

bibliographyOutput="C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Article/bibliographyarticlepackage.bib",

rgeos,

geosphere,

phytools,

geiger,

RPANDA,

caper,

neurobase,

misc3d,

phylolm,

nlme,

MCMCglmm,

coda,

strucchange

)

#Load environments

load("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/REnvironments/Data\_spatial\_primate.RData")

load("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/REnvironments/geography\_traits\_biogeobears.RData")

load("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/REnvironments/PGLSdiversification\_withautocorr.RData")

load("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/REnvironments/PGLSdirectionSelection.RData")

load("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/REnvironments/PGLSdiversificationAndSympatry.RData")

```

\captionsetup{list=no}

\newpage

```{r wordCount}

words <- RmdWords("Article.Rmd")

```

:::: {.graybox data-latex=""}

\*\*Abstract |\*\* Main hypotheses related to animal intelligence evolution highlight the role of conspecifics. Yet, space is a place often occupied by many species belonging to the same ecological guild. These sympatric heterospecifics can act as competitors (or cooperators) which stimulate cognition. However, they can also contribute to modifying and depleting the environment, making it poorly predictable and thereby alleviating the benefice of cognition. Considering brain size as a proxy for cognition, we used primates to test for the intertwine between species sympatry and cognition evolution and retraced the evolutionary history of several brain areas. We found that competitive evolutionary scenarios best predicted the evolution of brain areas related to interacting with the social or ecological environment, with a decrease of their relative size the higher the sympatry rate. Moreover, sympatry negatively affect primate diversification. Overall, this comparative study suggests that species sympatry, which likely convolutes the environment, contributes to shaping the selective pressure acting upon primate cognition and diversification.

\hfill

<!-- Reviewer possibles: Decasien, Powell, Barton, Kamilar, Harmon, Drury -->

\faTags \hspace{0.01in} \*\*Keywords:\*\* Brain size - Cognition - Competition - Co-occurrence - Diversification - Intelligence evolution - Primates - Sympatry

\faInfoCircle \hspace{0.01in} \*\*Word Count:\*\* `r words$num\_words` \newline

<!-- Character Count: `r words$num\_char` -->

::::

\newpage

```{r sampleCalculation, warning = FALSE, message = FALSE}

###Set working directory

setwd("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates")

repetition=2\*2\*2\*10#length(frugivoryThresholdVector)\*length(folivoryThresholdVector)\*length(geographicThresholdVector)\*randomSampling

checkSampleFruit <- rep(NA, times=repetition)

checkSampleLeaf <- rep(NA, times=repetition)

checkSampleRange <- rep(NA, times=repetition)

checkSampleBrain <- rep(NA, times=repetition)

checkSampleEQ <- rep(NA, times=repetition)

checkSampleNeocortex <- rep(NA, times=repetition)

checkSampleHippocampus <- rep(NA, times=repetition)

checkSampleCerebellum <- rep(NA, times=repetition)

checkSampleStriatum <- rep(NA, times=repetition)

checkSampleMOB <- rep(NA, times=repetition)

checkSampleRange <- rep(NA, times=repetition)

counter=0

for(a in 1:2){

for(b in 1:2){

for(c in 1:2){

for(d in 1:10){

counter=counter+1

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleFruit",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleFruit[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleLeaf",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleLeaf[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleRange",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleRange[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleBrain",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleBrain[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleEQ",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleEQ[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleNeocortex",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleNeocortex[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleHippocampus",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleHippocampus[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleCerebellum",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleCerebellum[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleStriatum",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleStriatum[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/Sample\_size/checkSampleMOB",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

checkSampleMOB[counter] <- toAdd[1]

}, error=function(e){

#Do nothing

}

)

}

}

}

}

checkSampleFruit <- unlist(checkSampleFruit)

checkSampleLeaf <- unlist(checkSampleLeaf)

checkSampleRange <- unlist(checkSampleRange)

checkSampleBrain <- unlist(checkSampleBrain)

checkSampleEQ <- unlist(checkSampleEQ)

checkSampleNeocortex <- unlist(checkSampleNeocortex)

checkSampleHippocampus <- unlist(checkSampleHippocampus)

checkSampleCerebellum <- unlist(checkSampleCerebellum)

checkSampleStriatum <- unlist(checkSampleStriatum)

checkSampleMOB <- unlist(checkSampleMOB)

#Min values

minFruit <- min(checkSampleFruit)

minLeaf <- min(checkSampleLeaf)

minRange <- min(checkSampleRange)

minBrain <- min(checkSampleBrain)

minEQ <- min(checkSampleEQ)

minNeocortex <- min(checkSampleNeocortex)

minHippocampus <- min(checkSampleHippocampus)

minCerebellum <- min(checkSampleCerebellum)

minStriatum <- min(checkSampleStriatum)

minMOB <- min(checkSampleMOB)

minAllAreas <- min(

minEQ,

minNeocortex,

minHippocampus,

minCerebellum,

minStriatum,

minMOB

)

#Max values

maxFruit <- max(checkSampleFruit)

maxLeaf <- max(checkSampleLeaf)

maxRange <- max(checkSampleRange)

maxBrain <- max(checkSampleBrain)

maxEQ <- max(checkSampleEQ)

maxNeocortex <- max(checkSampleNeocortex)

maxHippocampus <- max(checkSampleHippocampus)

maxCerebellum <- max(checkSampleCerebellum)

maxStriatum <- max(checkSampleStriatum)

maxMOB <- max(checkSampleMOB)

maxAllAreas <- max(

maxEQ,

maxNeocortex,

maxHippocampus,

maxCerebellum,

maxStriatum,

maxMOB

)

```

<!-- TC:endignore -->

# Introduction

<!-- On the road to brain size evolution, generally considered as an equivalent of cognition evolution, mysteries are plenty [@van2006some; @dunbar2017there]. Overall, the evolution of the brain is the consequence of constraints (e.g. energetic limitations, @navarrete2011energetics), but also of socio-ecological drivers promoting cognitive abilities [@gonzalez2018inference]. -->

<!-- Primates are pivotal species to study the evolution of cognition given the complexity of their socio-ecological environment and the inevitable implication for retracing human evolutionary history [@byrne2000evolution]. -->

\initial{\textcolor{black}{C}}ognition evolution results from the balance between socio-ecological drivers promoting cognitive abilities [@gonzalez2018inference] and physiological and energetic constraints [@navarrete2011energetics]. Primates are pivotal species for cognitive studies [@byrne2000evolution] because their cognition is thought to be promoted by interactions of individuals with conspecifics within the social unit [@byrne1994machiavellian; @dunbar2017there], among generations [@wilson1991molecular; @whiten2007evolution; @reader2002social; @herrmann2007humans; @tomasello2009cultural; @van2011social], between social units [@ashton2020interactions], or with the rest of their environment [@clutton1980primates; @milton1981distribution; @rosati2017foraging]. However, space is a place often occupied by many species belonging to the same ecological guild and we can predict that such interactions with sympatric heterospecifics are likely to strongly shape the evolution of cognition.

| Retracing the evolutionary history of cognitive abilities proves to be challenging because there is still no consensual measurement applicable across all species. Up to now, a raw approximation consists in considering brain size as a proxy for cognitive abilities (but see discussion in @logan2018beyond). Yet, the brain is a mosaic of areas cognitively specialized [@barton2000mosaic] and these areas can be differentially affected by species sympatry. For instance, the Striatum is stimulated during social interactions [@baez2013role]. Its size is thus expected to be positively influenced when contacts with other species increase, as in species performing mixed-species groups, which is nonetheless rather infrequent in primates (but see callitrichine primates in particular, @heymann2015unlike). The Striatum also underpins reward expectation and action, goal-directed behaviour and planning abilities [@johnson2007integrating] which is key when foraging. Other foraging-related areas, as the Main Olfactory Bulb (MOB) or the Hippocampus, home of a spatio-temporal memory [@burgess2002human], are also expected to be influenced by the ecological environment. Such areas are thought to be particularly important for frugivorous primates, as fruits are the archetype of a hard-to-find resource yet predictable [@janmaat2016spatio; @robirabotany] and larger areas might be more advantageous. Similarly, more generally-used areas, such as the Cerebellum [@koziol2014consensus; @sokolov2017cerebellum] and the Neocortex [@wiltgen2004new] underlying movement and/or general information processing and retention, could also be stimulated by the presence of other species movement traces, noises or odors to decode. [Il y a pas une aire pour laquelle on attend vraiement rien de particulier ? genre un controle négatif ? si oui le dire ici car sinon on a l’impression qu’on s’attend à un effet positif sur tout le monde….]. Therefore, sympatric heterospecifics can altogether stimulate cognition and promote larger sizes of the brain areas related to interacting with the social or ecological environment.

However, if changes induced by sympatric species with similar dietary habits increase the environmental unpredictability too much, sympatry can on the contrary change the selective pressure applying to cognitive abilities [@grove2013evolution: @robirainreview]. Indeed, due to the cost of maintaining the brain functional [@raichle2006brain], it is expected that the increase in environmental unpredictability only stimulates cognition up to a certain threshold (i.e. positive selection for bigger brain), before smaller brain sizes become more adaptive.

| Here, we investigated the intertwine between species sympatry and cognition using frugivorous primate as study example. To infer the effect of species interactions on brain size evolution within the frugivorous primates, we evaluated the support for competitive or non-competitive evolutionary scenarios in the context of sympatry and investigated the directionality of the selection upon brain size evolution in the context of sympatry. Finally, we tested for correlative pattern of the evolutionary success (species diversification) in primates with brain size or current sympatry context.

# Results

```{r calculationValueForFirstResults}

#Import data

summaryDataForPlot <- read\_delim("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/OutputEvolModel/Dataplot.txt","\t", escape\_double = FALSE, trim\_ws = TRUE)

summaryDataForPlot <- summaryDataForPlot[!is.na(summaryDataForPlot$geographicCode),]

# summaryDataForPlot <- summaryDataForPlot[summaryDataForPlot$DietaryGuild=="Fruit",]

summaryDataForPlot$Family <- Data\_powell2$MSW05\_Family[match(summaryDataForPlot$Species\_abbrv,Data\_powell2$Species\_abbrv)]#summaryData$Family[match(summaryDataForPlot$Species\_abbrv, summaryData$Species\_abbrv)]

summaryDataForPlot[is.na(summaryDataForPlot$Family ),c(1,2,3)]

#summaryDataForPlot$SpeciesForPhylogeny[which(is.na(summaryDataForPlot$Family))]

summaryDataForPlot$Family[summaryDataForPlot$Species\_abbrv=="Galagoides\_demi"]<-"Galagonidae"

summaryDataForPlot$Family[summaryDataForPlot$Species\_abbrv=="Macaca\_munz"]<-"Cercopithecidae"

summaryDataForPlot$Family[summaryDataForPlot$Species\_abbrv=="Microcebus\_mitt"]<-"Cheirogaleidae"

summaryDataForPlot$Family[summaryDataForPlot$Species\_abbrv=="Mirza\_zaza"]<-"Lorisidae"

summaryDataForPlot$Family[summaryDataForPlot$Species\_abbrv=="Nycticebus\_java"]<-"Galagonidae"

summaryDataForPlot$Family[summaryDataForPlot$Species\_abbrv=="Tarsius\_lari"]<-"Tarsiidae"

lemuriformes.v <-

c(

"Daubentoniidae",

"Lemuridae",

"Indriidae",

"Cheirogaleidae",

"Lepilemuridae"

)

platyrrhini.v <- c(

"Atelidae",

"Aotidae",

"Pitheciidae",

"Cebidae"

)

meanseMOBlemu <- round(c(

mean(

(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %in% lemuriformes.v],

na.rm=TRUE),

sd(

(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %in% lemuriformes.v],

na.rm=TRUE)/

sqrt(

length(

(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %in% lemuriformes.v &

!is.na(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass)]))

), digit=2)

meanseMOBother <- round(c(

mean(

(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %nin% lemuriformes.v],

na.rm=TRUE),

sd(

(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %nin% lemuriformes.v],

na.rm=TRUE)/

sqrt(

length(

(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %nin% lemuriformes.v &

!is.na(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass)]))

), digit=2)

meanseStriatumplaty <- round(c(

mean(

(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %in% platyrrhini.v],

na.rm=TRUE),

sd(

(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %in% platyrrhini.v],

na.rm=TRUE)/

sqrt(

length(

(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %in% platyrrhini.v &

!is.na(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass)]))

), digit=2)

meanseStriatumother <- round(c(

mean(

(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %nin% platyrrhini.v],

na.rm=TRUE),

sd(

(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %nin% platyrrhini.v],

na.rm=TRUE)/

sqrt(

length(

(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass)[summaryDataForPlot$Family %nin% platyrrhini.v &

!is.na(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass)]))

), digit=2)

# summaryDataForPlot[order(summaryDataForPlot$MOB/summaryDataForPlot$Bodymass, decreasing = TRUE),]

# summaryDataForPlot[order(summaryDataForPlot$Hippocampus/summaryDataForPlot$Bodymass, decreasing = TRUE),]

# summaryDataForPlot[order(summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass, decreasing = TRUE),]

```

We gathered a large database on the whole brain size, the size of the Cerebellum and the Neocortex, involved in daily information processing and retention, and the Main Olfactory Bulb (MOB), the Striatum, and the Hippocampus, those three latter being related to social and foraging-related tasks, containing between `r minAllAreas` to `r maxEQ` frugivorous primate species (depending on the brain area considered). It evidenced ample variations in brain size. For instance, the lemuriformes, that are known to prioritize smell compared to other primate species, have the largest relative MOB size (i.e. pondered by whole-body mass) in our data (Lemuriformes: mean $\pm$ SE = `r meanseMOBlemu[1]` $\pm$ `r meanseMOBlemu[2]`, other: `r meanseMOBother[1]` $\pm$ `r meanseMOBother[2]`, \@ref(fig:figbrain)). Similarly, platyrrhini, and callitrichine primates in particular, are known to form poly-specific associations [@heymann2000behavioural] and indeed show the highest relative size of the Striatum in our data (Platyrrhini: mean $\pm$ SE = `r meanseStriatumplaty[1]` $\pm$ `r meanseStriatumplaty[2]`, other: `r meanseStriatumother[1]` $\pm$ `r meanseStriatumother[2]`, \@ref(fig:figbrain)).

| We first reconstructed primate biogeography history (N$\_{species}$ = `r minRange`, @matzke2013probabilistic; @matzke2016stochastic) when considering `r length(areaName)` biogeographic areas (Figure \@ref(fig:figmap); @kamilar2009environmental) and their diet evolution (N$\_{species}$ = `r minFruit + minLeaf` to `r maxFruit + maxLeaf`; @bollback2006simmap). We then calculated the likelihoods of models considering the role of species interactions (matching competition MC models, @nuismer2015predicting; models of linear or exponential density dependence, DD$\_{lin}$and DD$\_{exp}$, of the evolutionary rate on the number of sympatric lineages; @drury2016estimating see [Models of trait evolution: does interspecific interactions shape brain size evolution?]) in the evolution of either the whole brain (using the encephalic quotient, EQ, as a proxy), or the size of the aforementioned specific brain areas relative to the whole-body mass (Figure. \@ref(fig:figbrain)) and compared their likelihood to those of simpler models assuming no effect of species interactions, like the Brownian Motion (BM), the Ornstein-Uhlenbeck process (OU) considering that traits are constrained around on optimal value (e.g. stabilizing selection; see @blomberg2020beyond for a review) or the Early-Burst model (EB, @blomberg2003testing), this latter allowing to check for a time-dependence of the evolutionary rate, hence emphasizing that, if any, the density effect is not an artefact due to time-dependence. Support for each model was evaluated using an information-theoretic framework [@burnham2002model] based on the weights of the Akaike Information Criterions corrected for small samples (AICc) when considering all six models (MC, DD$\_{lin}$, DD$\_{exp}$, BM, OU, EB, see [Models of trait evolution: does interspecific interactions shape brain size evolution?]).

| We found that non-competitive models were the most likely in describing the evolutionary history of the EQ, the Neocortex, and the Cerebellum (Figure \@ref(fig:figbrain) and \@ref(fig:figresultsevolution)), two areas specifically involved in movement and/or general information processing [@wiltgen2004new; @koziol2014consensus; @sokolov2017cerebellum] but also in memory consolidation for the Neocortex [@wiltgen2004new]. The fact that these biggest areas are best described by the Ornstein-Uhlenbeck process suggests a stabilization towards an optimal size and illustrates the trade-offs between costs and benefices of brain development [@isler2009expensive]. By contrast, competitive models were most supported in foraging-related and social areas, either specialised in spatio-temporal memory (Hippocampus, @burgess2002human) or planning abilities and reward assessment (Striatum, @baez2013role; @johnson2007integrating), while they tended to be so for one involved in sensory abilities (the main olfactory bulb, MOB; Figure \@ref(fig:figbrain) and \@ref(fig:figresultsevolution)). When density-dependent models were the best fit, the rate (\*r\*, Figure \@ref(fig:figresultsevolution)) suggested an acceleration of the evolutionary tempo together with increased lineage diversity for the Hippocampus and the Striatum, but a slowdown for the MOB.

```{r samplecooccurrence}

minNspeciesCoocc <- round(min(dataRangePrimate$Number\_species\_cooccurrence, na.rm=TRUE), digit=2)

maxNspeciesCoocc <- round(max(dataRangePrimate$Number\_species\_cooccurrence, na.rm=TRUE), digit=2)

meanNspeciesCoocc <- round(mean(dataRangePrimate$Number\_species\_cooccurrence, na.rm=TRUE), digit=2)

seNspeciesCoocc <- round(sd(dataRangePrimate$Number\_species\_cooccurrence, na.rm=TRUE)/sqrt(length(dataRangePrimate$Number\_species\_cooccurrence[!is.na(dataRangePrimate$Number\_species\_cooccurrence)])), digit=2)

whichSpeciesMinNCoocc <- paste(sort(dataRangePrimate$Species[which(round(dataRangePrimate$Number\_species\_cooccurrence, digit=2)==minNspeciesCoocc)]), collapse=", ")

whichSpeciesMaxNCoocc <- paste(sort(dataRangePrimate$Species[which(round(dataRangePrimate$Number\_species\_cooccurrence, digit=2)==maxNspeciesCoocc)]), collapse=", ")

whichSpeciesMinNCoocc <- gsub("\_", " ", whichSpeciesMinNCoocc)

whichSpeciesMaxNCoocc <- gsub("\_", " ", whichSpeciesMaxNCoocc)

minOverlapSpecies <- round(min(dataRangePrimate$Overlap\_average, na.rm=TRUE), digit=2)

maxOverlapSpecies <- round(max(dataRangePrimate$Overlap\_average, na.rm=TRUE), digit=2)

meanOverlap <- round(mean(dataRangePrimate$Overlap\_average, na.rm=TRUE), digit=2)

seOverlap <- round(sd(dataRangePrimate$Overlap\_average, na.rm=TRUE)/sqrt(length(dataRangePrimate$Overlap\_average[!is.na(dataRangePrimate$Overlap\_average)])), digit=2)

whichSpeciesMinOverlap <- paste(dataRangePrimate$Species[which(round(dataRangePrimate$Overlap\_average, digit=2)==minOverlapSpecies)], collapse=", ")

whichSpeciesMaxOverlap <- paste(dataRangePrimate$Species[which(round(dataRangePrimate$Overlap\_average, digit=2)==maxOverlapSpecies)], collapse=", ")

whichSpeciesMinOverlap <- gsub("\_", " ", whichSpeciesMinOverlap)

whichSpeciesMaxOverlap <- gsub("\_", " ", whichSpeciesMaxOverlap)

```

```{r estimateRegGradient, error=TRUE, include=TRUE}

#Hippocampus

model <- get(paste("modelBrain", traitName[3], sep="\_"))

estimateGradientHippocampus <- textEstOneVar(summary(model)$coefficients[2,c(1,4,5)])

testGradientHippocampus <- textTestOneVar(summary(model)$coefficients[2,c(3,6)], statistics="")

#Striatum

model <- get(paste("modelBrain", traitName[6], sep="\_"))

estimateGradientStriatum <- textEstOneVar(summary(model)$coefficients[2,c(1,4,5)])

testGradientStriatum <- textTestOneVar(summary(model)$coefficients[2,c(3,6)], statistics="")

#MOB

model <- get(paste("modelBrain", traitName[7], sep="\_"))

estimateGradientMOB <- textEstOneVar(summary(model)$coefficients[2,c(1,4,5)])

testGradientMOB <- textTestOneVar(summary(model)$coefficients[2,c(3,6)], statistics="")

```

| Next, to understand the directionality of the selection gradient shaped by sympatry (i.e. selection for “bigger” or “smaller” brain the more species), we fitted phylogenetic regressions (see [Phylogenetic regressions] a)). For these linear regressions, the predicted variable was the relative brain size values of the different areas. We considered the average surface of the frugivorous species range that was overlapped by other sympatric frugivorous species, as well as the number of such sympatric frugivorous species across their entire distribution range as covariates based on IUCN date [@IUCN]. On average ($\pm$ SE), the considered primate species had `r meanOverlap\*100`% of their range overlapping with other species ($\pm$ `r seOverlap\*100`). That ranged from `r minOverlapSpecies\*100`% of overlap (\*`r whichSpeciesMinOverlap`\*), to `r maxOverlapSpecies\*100`% of overlap (\*`r whichSpeciesMaxOverlap`\*). In terms of distribution range, the considered primate species co-occurred on average with `r meanNspeciesCoocc` other primate species ($\pm$ `r seNspeciesCoocc`), ranging from `r minNspeciesCoocc` other species (\*`r whichSpeciesMinNCoocc`\*), to `r maxNspeciesCoocc` species (\*`r whichSpeciesMaxNCoocc`\*). The number of sympatric species never influenced significantly the relative size of the brain or other specific areas (Table \@ref(tab:tableRegGradient)). Conversely, we found that the percentage of range shared with other species correlated with the relative size of areas that were better fit with competitive models: The Hippocampus, and the Striatum (Hippocampus: $t$`r testGradientHippocampus`; Striatum: $t$`r testGradientStriatum`). The correlations were all negative (Hippocampus: `r estimateGradientHippocampus`; Striatum: `r estimateGradientStriatum`), which means that higher species overlap associates with lower relative size, insensitive to data and phylogenetic uncertainty (Table \@ref(tab:tabledfsensitivity), Figure \@ref(fig:forestPlot)). Thus, it suggests that sympatric species are either subject to a positive selection towards smaller brain, or to a less intense selection for advanced cognitive abilities compared with isolated species.

```{r estimateRegDiversification, error=TRUE, include=TRUE}

#print(traitName)

# increaseHippocampus <- round(exp(summary(model)$coefficients[2,c(1)]), digit=2)

# averageDivRateHippocampus <- round(exp(summary(model)$coefficients[1,c(1)]), digit=2)

# estimateHippocampus <- textEstOneVar(summary(model)$coefficients[2,c(1,4,5)])

# testHippocampus <- textTestOneVar(summary(model)$coefficients[2,c(3,6)], statistics="")

#

# #Neocortex

# model <- get(paste("modelBrainDiversification", traitName[4], sep="\_"))

#

# increaseNeocortex <- round(exp(summary(model)$coefficients[2,c(1)]), digit=2)

# averageDivRateNeocortex <- round(exp(summary(model)$coefficients[1,c(1)]), digit=2)

# estimateNeocortex <- textEstOneVar(summary(model)$coefficients[2,c(1,4,5)])

# testNeocortex <- textTestOneVar(summary(model)$coefficients[2,c(3,6)], statistics="")

#

# #Striatum

# model <- get(paste("modelBrainDiversification", traitName[6], sep="\_"))

#

# increaseStriatum <- round(exp(summary(model)$coefficients[2,c(1)]), digit=2)

# averageDivRateStriatum <- round(exp(summary(model)$coefficients[1,c(1)]), digit=2)

# estimateStriatum <- textEstOneVar(summary(model)$coefficients[2,c(1,4,5)])

# testStriatum <- textTestOneVar(summary(model)$coefficients[2,c(3,6)], statistics="")

#print(c(increaseStriatum, averageDivRateStriatum, estimateStriatum, testStriatum))

```

```{r determineBoomTime}

###-----

## Diversification evolution over time

fraction.v <- c(60, 70, 80, 90, 95)

table\_MAPS\_rates\_time <- c()

for (f in fraction.v){

table\_MAPS\_rates\_time\_transitory <- read.table(paste0("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Scripts/Analysis3\_diversification/diversification/MAPS\_speciation\_rates\_trought\_time\_ClaDS2\_tree\_primate\_complete\_f",f,".csv"),sep=";",header=T)###Hahah nice English Benoit, trought ;)

table\_MAPS\_rates\_time\_transitory$f <- f

table\_MAPS\_rates\_time <- rbind(table\_MAPS\_rates\_time, table\_MAPS\_rates\_time\_transitory)

}

aggregatedSpeciationTime.mean <- aggregate(table\_MAPS\_rates\_time$speciation\_rates, by = list(table\_MAPS\_rates\_time$time), FUN = mean)

aggregatedSpeciationTime.sd <- aggregate(table\_MAPS\_rates\_time$speciation\_rates, by = list(table\_MAPS\_rates\_time$time), FUN = sd)

colnames(aggregatedSpeciationTime.mean) <- c("Time", "Diversification")

colnames(aggregatedSpeciationTime.sd) <- c("Time", "Diversification")

# bp.resp <- breakpoints(Diversification ~ Time, data=aggregatedSpeciationTime.mean)

# summary(bp.resp)

# ## the BIC

# plot(bp.resp)

# breakpoints(bp.resp)

#Suggest that 4 is better from stats, visually 2

bp.resp <- breakpoints(Diversification ~ Time, data=aggregatedSpeciationTime.mean, breaks=2)

ci.resp <- confint(bp.resp)

dates <- as.data.frame(breakdates(ci.resp))

dateFirstRupt <- as.numeric(round((1-dates[1,])\*(max(aggregatedSpeciationTime.mean$Time) - min(aggregatedSpeciationTime.mean$Time)), digit=2))

dateSecondRupt <- as.numeric(round((1-dates[2,])\*(max(aggregatedSpeciationTime.mean$Time) - min(aggregatedSpeciationTime.mean$Time)), digit=2))

```

| In addition, we investigated the evolutionary consequences of brain evolution and sympatry by evaluating whether brain sizes and sympatry level correlated with species diversification rates (i.e. speciation minus extinction rates), by using lineage-specific birth-death models of species diversification [@maliet2019model; @maliet2020fast]. Overall, species diversification, estimated based on molecular phylogeny increased over time (Figure \@ref(fig:figdiversificationTime)), particularly in the early and late Miocene, around `r dateFirstRupt[2]` (CI95% = `r paste("[",dateFirstRupt[3],",",dateFirstRupt[1],"]", sep="")`) and `r dateSecondRupt[2]` (CI95% = `r paste("[",dateSecondRupt[3],",",dateSecondRupt[1],"]", sep="")`) Myr ago (Figure \@ref(fig:figdiversificationTime)). When accounting for phylogenetic dependence, no significant relationship between diversification rate and the relative size of brain areas was found (Table \@ref(tab:tableRegDiversification), Figure \@ref(fig:forestPlot); see robustness in Table \@ref(tab:tabledfsensitivity2)). Given the context dependence of the direction of selection (towards bigger sizes in the absence of other species, smaller sizes otherwise), there is no surprise that we do not observe a correlation between evolutionary success (approximated by diversification rate) and the three studied areas affected by inter-species sympatry. Surprisingly however, we found no positive association between the EQ and the diversification rate, nor of that latter with the relative size of the Cerebellum or the Neocortex, two areas insensitive to inter-species sympatry too. This is puzzling because this contradicts a recent study [@melchionna2020macroevolutionary]. The visual inspection of the regressions however clearly evidenced a positive trend for the EQ and the Neocortex if discarding phylogenetic non-independence (Figure \@ref(fig:figRegressionDiversification)). In fact, a sudden encephalisation in primates is clearly associated to a limited number of closely-related species [@decasien2017primate; @melchionna2020macroevolutionary]. This clearly limits the statistical power of our phylogenetically corrected analyses as we cannot decipher whether the connivance between brain size and evolutionary success results from a true biological link or appeared by chance. A positive association between brain size and diversification was also found in birds [@sayol2019larger] given that bigger brains act as a buffer to environmental challenge, thus boosting individuals' fitness [@sol2007big]. This means that, despite what we found here, the positive association between brain size and evolutionary success evidenced in @melchionna2020macroevolutionary remains a likely possibility. Finally, although diversification was uncorrelated with brain size in frugivorous primates, it was influenced by the sympatry context. In particular, phylogenetic regressions highlighted a negative effect of the number of sympatric species on the diversification rate (`r textEstOneVar(as.numcharac(results.df\_diversificationAndSympatry[4,2:4]))`, `r textTestOneVar(as.numcharac(results.df\_diversificationAndSympatry[4,5:6]), statistics="t")`, Table \@ref(tab:tableRegDiversificationAndSympatry), Figure \@ref(fig:forestPlot)). In other words, the higher the number of sympatric species, the lower the diversification, a trend that is frequently suggested in many tetrapod clades, best fit by density-dependent diversification models [REF].

an inverted pattern compared to what is found in New World passerines [@kennedy2017historical].

# Discussion

The bigger is not necessarily the better. The size of the brain is subject to a compromise between the energy it incurs, and the increase of fitness it allows. This is clearly emphasized by the fact that the biggest areas, the Cerebellum and the Neocortex, as well as the whole brain (EQ) were best described by the Orstein-Uhlenbeck process, what might suggest a stabilisation towards an optimal size as a result of an equilibrium between costs and benefices. As the brain is regionally specialised [@barton2000mosaic], brain regions could be under different selective pressures as suggested by the differences we highlighted in some brain areas (e.g. MOB or Striatum) across different primate genus. Here, we further show that sympatry is one factor that affects the selective regime under which each brain region evolves: Although the brain as a whole was insensitive to species sympatry, this latter nonetheless induced a change in the relative size of the Hippocampus, the Striatum, and potentially in the MOB of frugivorous primates. These areas are involved in individual-based and social-based information processing, pinpointing that the two components might be under selection in primates [@decasien2017primate; @powell2017re; @gonzalez2018inference]. Overall, the relative sizes of foraging-related brain areas were negatively associated with one index of sympatry intensity (the range overlap with other species) and might thus be a consequence of direct or indirect interspecific interactions. In general, competition is the first-thought mechanism to describe community structures [e.g. @rocha2015role]. In primates, it is rare to observe direct competition between species, which should translate into site exclusion. In fact, inter-species site exclusion in this taxon has only been observed in gibbons [@suwanvecho2012interspecific]. For species with overlapping diet, indirect competition for resources might happen. It has been shown that individuals tend to use social or environmental cues depending on their reliability [@rafacz2003environmental; @dunlap2016foraging]. The utilization of these cues should promote cognition until reaching a certain range of environmental complexity [@grove2013evolution; @robirainreview]. Our results could thus suggest that species sympatry may trigger an important decrease in cues reliability, for instance due to higher and unforeseen depletion rate. That would explain a relaxed positive selection for advance cognitive abilities, and conversely, given the cost of brain maintenance [@raichle2006brain], would induce a negative selection towards the sizes of areas involved in foraging-related activities like the Hippocampus, the MOB and the Striatum. Not only was brain evolution affected by sympatry, but sympatry (the maximum number of co-occurring species only) induced a slowdown in diversification, which contrasts with the picture offered by passerine birds at finer resolution [@kennedy2017historical]. Density-dependence within clade is often thought to decrease diversification rate [@moen2014does]. In particular, species competing for resource are thought to contribute to limiting competitors' range [@price2009evolutionarily], hence to constrain population size and diversification rate [@pigot2013species]. Given the large resolution and low overlap threshold to consider sympatry considered here (i.e. overlap of `r geographicThresholdVector[1]\*100`% of the distribution area), this competitive view fits the observed influence of sympatry on frugivorous primate diversification.

| The use of brain size as a proxy for cognition is a central debate with no optimal solution (see grounded criticism from @deaner2000comparative; @healy2007critique; @logan2018beyond). The current flourishment of consortia, allowing for much more detailed and standardized anatomical measurements (e.g. in primates: @milham2018open), or with standardized behaviourally explicit comparisons (e.g. on captive [@many2019establishing] or wild [@janmaat2021using] primates), might alleviate biases stemming from brain size analysis, but this will take time to generate large-enough datasets. In the meanwhile, brain size is a proxy much appreciated in practice, because of its easy accessibility for a “large” number of species. Here, we showed that species sympatry can influence the evolutionary scenario that applies to animals' brain, but the proximate mechanisms at play remain to be elucidated. Finally, it is very likely that any hypothesis on cognition evolution, generally discussed within species, could be broadened to a between-species context: Polyspecific social associations do exist [@porter2001benefits], as well as inter-species territory defense [@drury2020competition; @losin2016ecological] or imitation and copying [@persson2018spontaneous; @pepperberg2002allospecific]. As Alice said “It's a great huge game of chess that's being played—all over the world” (@Carroll, Chapter II) and all individuals are just pieces to play with or against, no matter the species.

<!-- TC:ignore -->

\newpage

# Methods

Data processing, analyses, and plots were computed with R software [`r extractRversion()`, @Rsoftware]. Used codes and data are freely available at [https://github.com/benjaminrobira/Temporal\_memory\_and\_foraging\_efficiency](https://github.com/benjaminrobira/Temporal\_memory\_and\_foraging\_efficiency). Note that in all these analyses, we discarded \*Homo sapiens\* and \*Macaca sylvanus\*, this latter being too geographically isolated. A summary of available data per species is presented in Appendix Figure \@ref(fig:figmap).

## Data Collection

### Phylogeny

We used a block of chronogram trees of the primate taxon of the 10kTrees project (downloaded on the 11/05/2021, version 3), as well as a consensus tree of 1000 trees for the subsequent phylogenetic analyses. The trees contain `r length(phylo\_init$tip.label)` primate species.

### Trait data

Brain data were obtained from @decasien2019primate for whole brain and all mentioned other parts (Cerebellum, Hippocampus, Main Olfactory Bulb (MOB), Neocortex, Striatum), @powell2017re and @powell2019maternal for whole brain, Cerebellum and Neocortex size, @todorov2019primate for Hippocampus and Neocortex size, @grueter2015home for the whole brain size and @navarrete2018primate for the whole brain, Cerebellum, Hippocampus and Striatum size. They were freely available in the main manuscript or supplementary materials. When a species was represented multiple times within dataset, we obtained a unique attribute by averaging it. From the global endocranial brain volume, we obtained the Encephalization Quotient (EQ, N$\_{EQ,max}$ = `r maxEQ`) as follows [@decasien2017primate]

\hfill

\begin{center}

$\mathrm{EQ}=1.036 \times \mathrm{Brainvolume}/ (0.085 \times \mathrm{Body mass}^{0.775})$

\end{center}

\hfill

with the brain volume in cm$^{3}$, 1.036 g/cm$^{3}$ being the assumed homogeneous brain density, and the body mass in g. EQ indicates whether the brain size ranges above (> 1) or below (< 1) expected given the body mass. Body mass was obtained from @decasien2017primate, @powell2017re, @grueter2015home and @pearce2013space.

The sub-parts of the brain were chosen because they were involved in immediate sensory information processing (MOB, N$\_{MOB,max}$ = `r maxMOB`), in movement and/or general information processing and retention (Neocortex, N$\_{Neocortex,max}$ = `r maxNeocortex`, @wiltgen2004new; Cerebellum, N$\_{Cerebellum,max}$ = `r maxCerebellum`, @koziol2014consensus; @sokolov2017cerebellum), short-term working memory and long-term spatio-temporal memory (Hippocampus, N$\_{Hippocampus,max}$ = `r maxHippocampus`, @burgess2002human). The Striatum (N$\_{Striatum,max}$ = `r maxStriatum`) supports information processing during social interaction, reward assessment, planning or goal-oriented behaviours [@baez2013role; @johnson2007integrating]. To investigate their evolutionary history, we used the ratio between their volume and body mass, so as to maximize comparability. As such, the use of specific region size relatively to the body mass and not raw size depicts the evolution of cognitive abilities in terms of allocation rather than abilities per se (but see discussion in @deaner2000comparative). Percentage of frugivory and/or folivory was obtained based on freely available dataset from @decasien2017primate and @powell2017re for the frugivory and folivory rate, or @willems2013collective for the folivory rate.

### Ranging Data

Current geographic (maximal possible) range of each primate species was assessed using ranging maps provided by the IUCN red list [@IUCN]. Ranging data were available for `r nrow(matrixRangingSensitivity[!is.na(matrixRangingSensitivity[,1]),])` species among the `r length(phylo\_init$tip.label)` represented in the 10kTrees primate phylogeny.

## Primate species sympatry

One to multiple large-scale geographic areas were assigned to each species as soon as the species current distribution range overlapped in surface at `r geographicThresholdVector[1]\*100` (low threshold) or `r geographicThresholdVector[2]\*100`% (high threshold; the maximum was chosen to `r geographicThresholdVector[2]\*100`% because on present data, a species could occupy as far as three areas, Figure \@ref(fig:figmap)). Overlap was calculated with the “gIntersection” function from the \*rgeos\* package [@rgeos] applied to Mercator-projected data to get the overlap contour, and the “area” function from the \*geosphere\* package [@geosphere], applied directly on unprojected longitudinal-latitudinal data for area size calculation. Based on the structure (i.e. number of species and their phylogenetic relationship) of primate communities at different field sites, @kamilar2009environmental determined clusters of sites with highly similar community structures that were shaped by both the environment geography and climatic correlates. We used this classification and manually mapped the geographic areas using Google earth professional (v7.3.3). These geographic areas are represented in Figure \@ref(fig:figmap) and correspond to Central America, the North and the South of South America respectively, West Africa, Central Africa, and East/South Africa, East and West of Madagascar respectively, West Asia, Central/East Asia, South Asia and the Asian Islands. The chosen scale for the areas is large because (i) retracing history of a large number of areas necessitates considerable computational means. In addition, this drastically increases computational time of phylogenetic model of brain trait evolution too. Furthermore (ii), all species and particularly primate species suffer(ed) from recent extinction [@pavoine2019mammal], with reduction of ranging areas at an unprecedented speed rate. Finer geographic characterization would therefore give too much weight to such anthropogenic effect that recently altered species distribution (e.g. evidenced on the North American fauna in @pineda2021mammal).

| We retraced the history of the lineage ranges based on current observations of species range using the \*BioGeoBEARS\* package [@matzke2013probabilistic], using the biogeographic stochastic mapping algorithm [@matzke2016stochastic]. We fitted non-time-stratified dispersal-extinction-cladogenesis (DEC) models specifically suiting analyses of range data since it accounts for spatially explicit processes of cladogenetic and anagenetic events (see @matzke2013probabilistic for further details on these events). To reconstruct the evolution of species range, we fixed the maximum numbers of areas that could be occupied by a lineage at one time to three areas. A too high number of areas that can be occupied simultaneously drastically increases computational time. Here, we therefore chose that a species can at most occupy three areas since it offers the possibility to occupy a complete mainland continent. Finally, because these history reconstructions are likely to vary, for each run of DEC models (considering both possible overlaps to consider species presence), we obtained `r numberSimulations` stochastic maps that were all used in subsequent phylogenetic model fits (see [Phylogenetic models]) to account for uncertainty of these ancestral range estimations (see [Models of trait evolution: does interspecific interactions shape brain size evolution?] (b)).

## Dietary guild

We classified species as either “frugivorous” or “folivorous” based on the availability of frugivorous rate and folivorous rate, prioritizing fruvigory over folivory. First, a species would be classified as frugivorous if the frugivory rate was at least above `r frugivoryThresholdVector[1]` (low threshold) or `r frugivoryThresholdVector[2]`% (high threshold). If this was not the case, or frugivory rate was unavailable, a species could be classified as folivorous if the folivory rate was at least above `r folivoryThresholdVector[1]` (low threshold) or `r folivoryThresholdVector[2]`% (high threshold). Otherwise, @decasien2017primate gave a binary classification of diet, species being categorized as frugivorous or folivorous, partly based on anatomical criteria. Whenever the rate was not available, we referred to this classification. In any other cases, the species was discarded.

```{r transitionMatrix}

transitionMatrix <- matrix(NA, nrow=repetition, ncol=2)

for(a in 1:2){

for(b in 1:2){

for(c in 1:2){

for(d in 1:10){

start=which(is.na(transitionMatrix[,1]))[1]

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/OutputEvolModel/Output\_simmap\_transition",a,"\_",b,"\_",c,"\_",d, ".txt", sep=""))

transitionMatrix[start,1] <- toAdd[1,1]

transitionMatrix[start,2] <- toAdd[2,1]

}, error=function(e){

#Do nothing

}

)

}

}

}

}

minProba.v <- apply(abs(transitionMatrix), 2, min)

maxProba.v <- apply(abs(transitionMatrix), 2, max)

```

| Frugivory rate was prioritized over folivory because we considered that since fruits are a highly palatable food source, it would be the key item that drives the foraging strategy (and associate consequence(s) on brain selection), even if less consumed. Additionally, to consider frugivory, we used a lower rate than for folivory for two reasons. First, such static rate does not reflect potential seasonality in fruit eating (e.g. @masi2009western), which is generally shorter, hence a lower overall frugivory rate. Second, frugivory rate is likely to be underestimated in part because primates generally spend more time feeding on leaves than fruits, while rates are often based on relative feeding time, or observation frequency at the individual or group unit of feeding events. Finally, the methodology to obtain this rate could additionally vary (e.g. in addition to the two aforementioned estimations, one could also rely on the proportion of species targeted for their fruits/leaves). For all these reasons, we used two threshold levels (low, `r frugivoryThresholdVector[1]`%, or high, `r frugivoryThresholdVector[2]`%) to classify a species as frugivorous, as well as two threshold levels (low, `r folivoryThresholdVector[1]`%, or high, `r folivoryThresholdVector[2]`%) to classify a species as folivorous.

| Considering diet as a binary variable (frugivory versus folivory), we retraced the evolutionary history of such discrete traits based on a continuous Markovian process (extended Mk models) and relying on a Bayesian approach [@bollback2006simmap], using the “simmap” function of the \*phytools\* package [@phytools] and internally estimating the prior probability of trait (i.e. at the root) but with no prior on the transition matrix. Again, the obtained character history is in no case certain. Therefore, for each run, we obtained `r numberSimulations` stochastic character maps that were used in subsequent phylogenetic model fits [Phylogenetic models] to account for uncertainty of these ancestral diet estimations (see [Phylogenetic models, Models of trait evolution: does interspecific interactions shape brain size evolution?] (b)).

## Phylogenetic models

### Models of trait evolution: does interspecific interactions shape brain size evolution?

\hfill

(a) Fitting models of trait evolution

\hfill

We focused only on frugivorous primates, because sample size was otherwise insufficient, and fitted phylogenetic models of EQ - or relative size of a specific brain area – evolution with and without species competitions. Models were fitted on different sample sizes due to non-availability of some data for some traits. Specifically, models using EQ included `r minEQ` to `r maxEQ` frugivorous species. Other models included more reduced sample sizes (in species number): Striatum (`r minStriatum` to `r maxStriatum`), MOB (`r minMOB` to `r maxMOB`), Neocortex (`r minNeocortex` to `r maxNeocortex`), Hippocampus (`r minHippocampus` to `r maxHippocampus`), Cerebellum (`r minCerebellum` to `r maxCerebellum`). Nonetheless, for a given set of models (i.e. within brain area), the sample was strictly identical, allowing within set comparison. Prior fitting, trait parameters were log-transformed to reach more symmetrical distributions. Models without competition, Brownian Motion (i.e. BM), Orstein-Uhlenbeck process (i.e. OU, model with stabilizing selection), or Early-Burst model (i.e. EB, for assessing a time-dependence of the evolutionary rate) were fitted using the “fitContinuous” function from the \*geiger\* package [@geiger3; @geiger5]. Using the evolutionary history of species distribution (see [Primate species sympatry]) and of diet (see [Dietary guild]), we fitted competitive models using the “fit\_t\_comp” function from the \*RPANDA\* package [@RPANDA]. These competitive models notably account for interaction matrices that are built on the evolutionary history of species sympatry and diet. These interaction matrices retrace, along the phylogenetic tree, which frugivorous lineages were present within the same geographic areas (see @drury2016estimating). We fitted three different competitive models. The matching competition model (MC) may consider divergence of traits of co-occurring lineages from a same dietary guild due to repulsion of traits (character displacement) [@drury2016estimating]. Here, that would mean that co-occurring species would tend to have either lower or higher EQ or relative brain size. Otherwise, we modelled trait evolution accounting for linear (DD$\_{lin}$) or exponential (DD$\_{exp}$) density-dependence [@drury2016estimating; @weir2013diversity]. Density-dependence means that the evolutionary rate $\lambda$ varies either positively or negatively as a function $f$ of the number of co-occurring lineages sharing the same diet such as

\begin{center}

\hfill

$f\_{lin}(\lambda)=\lambda\_{0} + rl$

$f\_{exp}(\lambda)=\lambda\_{0}\exp(rl)$

\hfill

\end{center}

where $\lambda\_{0}$ corresponds to the value of the initial ancestor, $l$ indicates the number of lineages, $r$ allows for modelling the speed and direction of the dependency to lineage number ($r>0$ leads to an increase of trait changes, while $r<0$ leads to a decline of the trait changes). All these models were repeated `r numberSimulations` times, using `r randomSampling` different combination of the evolutionary history of ranging and diet. They were then compared within an information-theoretic framework [@burnham2002model] based on the weight of Akaike Information Criterion corrected for small samples (AICc) when considering all six models (MC, DD$\_{lin}$, DD$\_{exp}$, BM, OU, EB). The model weight then depicts the probability that it best describes the observed evolutionary pattern among the tested models.

\hfill

```{r prepfigevolution}

###Set working directory

setwd("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates")

summaryBrainFrugivory <- as.data.frame(matrix(NA, nrow=10\*(repetition+1), ncol=53))

summaryEQFrugivory <- as.data.frame(matrix(NA, nrow=10\*(repetition+1), ncol=53))

summaryNeocortexFrugivory <- as.data.frame(matrix(NA, nrow=10\*(repetition+1), ncol=53))

summaryHippocampusFrugivory <- as.data.frame(matrix(NA, nrow=10\*(repetition+1), ncol=53))

summaryCerebellumFrugivory <- as.data.frame(matrix(NA, nrow=10\*(repetition+1), ncol=53))

summaryStriatumFrugivory <- as.data.frame(matrix(NA, nrow=10\*(repetition+1), ncol=53))

summaryMOBFrugivory <- as.data.frame(matrix(NA, nrow=10\*(repetition+1), ncol=53))

counter=0

start=counter

end=counter

numberSimulations=10

totModelsWorked=c(0,0,0,0,0,0,0)

for(a in 1:2){

for(b in 1:2){

for(c in 1:2){

for(d in 1:10){

counter=end+1

start=counter

end=counter + numberSimulations - 1

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/OutputEvolModel/Output\_evolutionary\_history\_BrainBodymassRaw",a,"\_",b,"\_",c,"\_",d,".txt", sep=""))

summaryBrainFrugivory[start:end,] <- as.data.frame(toAdd)

totModelsWorked[1]=totModelsWorked[1]+1

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/OutputEvolModel/Output\_evolutionary\_history\_EQ",a,"\_",b,"\_",c,"\_",d,".txt", sep=""))

summaryEQFrugivory[start:end,] <- toAdd

totModelsWorked[2]=totModelsWorked[2]+1

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/OutputEvolModel/Output\_evolutionary\_history\_NeocortexBodymassRaw",a,"\_",b,"\_",c,"\_",d,".txt", sep=""))

summaryNeocortexFrugivory[start:end,] <- toAdd

totModelsWorked[3]=totModelsWorked[3]+1

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/OutputEvolModel/Output\_evolutionary\_history\_HippocampusBodymassRaw",a,"\_",b,"\_",c,"\_",d,".txt", sep=""))

summaryHippocampusFrugivory[start:end,] <- toAdd

totModelsWorked[4]=totModelsWorked[4]+1

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/OutputEvolModel/Output\_evolutionary\_history\_CerebellumBodymassRaw",a,"\_",b,"\_",c,"\_",d,".txt", sep=""))

summaryCerebellumFrugivory[start:end,] <- toAdd

totModelsWorked[5]=totModelsWorked[5]+1

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/OutputEvolModel/Output\_evolutionary\_history\_StriatumBodymassRaw",a,"\_",b,"\_",c,"\_",d,".txt", sep=""))

summaryStriatumFrugivory[start:end,] <- toAdd

totModelsWorked[6]=totModelsWorked[6]+1

}, error=function(e){

#Do nothing

}

)

tryCatch(

{toAdd <- read.delim(paste("Processed\_data/OutputEvolModel/Output\_evolutionary\_history\_MOBBodymassRaw",a,"\_",b,"\_",c,"\_",d,".txt", sep=""))

summaryMOBFrugivory[start:end,] <- toAdd

totModelsWorked[7]=totModelsWorked[7]+1

}, error=function(e){

#Do nothing

}

)

}

}

}

}

summaryBrainFrugivory <- summaryBrainFrugivory[!is.na(summaryBrainFrugivory[,1]),]

summaryEQFrugivory <- summaryEQFrugivory[!is.na(summaryEQFrugivory[,1]),]

summaryNeocortexFrugivory <- summaryNeocortexFrugivory[!is.na(summaryNeocortexFrugivory[,1]),]

summaryHippocampusFrugivory <- summaryHippocampusFrugivory[!is.na(summaryHippocampusFrugivory[,1]),]

summaryCerebellumFrugivory <- summaryCerebellumFrugivory[!is.na(summaryCerebellumFrugivory[,1]),]

summaryStriatumFrugivory <- summaryStriatumFrugivory[!is.na(summaryStriatumFrugivory[,1]),]

summaryMOBFrugivory <- summaryMOBFrugivory[!is.na(summaryMOBFrugivory[,1]),]

colnames(summaryBrainFrugivory) <- colnames(toAdd)

colnames(summaryEQFrugivory) <- colnames(toAdd)

colnames(summaryNeocortexFrugivory) <- colnames(toAdd)

colnames(summaryHippocampusFrugivory) <- colnames(toAdd)

colnames(summaryCerebellumFrugivory) <- colnames(toAdd)

colnames(summaryStriatumFrugivory) <- colnames(toAdd)

colnames(summaryMOBFrugivory) <- colnames(toAdd)

##----

colNum <-c("darkgrey", brewer.pal(n = 5, name = "Set1")[1:5])

models <- c("BM", "OU", "EB", "MC", expression(DD[italic(lin)]), expression(DD[italic(exp)]))

colourModels <- brewer.pal(n = 6, name = "Set1")

```

(b) Dealing with data uncertainty and parameter sensitivity

\hfill

In this analysis, uncertainty can stem from two sources. First, the true phylogeny is never known with certainty, and is estimated through Bayesian inference, thus we used the consensus tree from the 10kTrees project, which averages the phylogeny among 1000 possible estimated trees, given that running the models on several trees was too computationally demanding.

| Similarly, the estimated evolutionary history of the diet and ranging might vary as well. Second, for each species, trait estimates could vary slightly among datasets (see Appendix Figure \@ref(fig:figvariabilitydata)). Particularly, although correlations seem good enough, it existed a variation in absolute measurement (Appendix Figure \@ref(fig:figvariabilitydata)), while, in order to increase the overall number of species, trait values were not mandatorily from a single dataset. In addition, this study is based on several arbitrary thresholds, namely (i) to assess species sympatry (see Appendix Figure \@ref(fig:figcomparison)) and (ii) to assess the species dietary guild (see Appendix Figure \@ref(fig:figvariabilitydata)) which can cause sensitivity of the results to the chosen parameters. To account for these three sources of variability we refitted several times the six models of trait evolution (BM, OU, EB, MC, DD$\_{lin}$ and DD$\_{exp}$) with (1) random samples of the dietary and brain traits in case of multiple values available (i.e. equal probability for each possible value to be selected), (2) used the low or high threshold for assessing frugivory, folivory and geographic sympatry, and (3) various biogeography and dietary evolutionary history reconstructions.

| Eventually, it means that the results for each model represent the average of `r numberSimulations` (uncertainty on diet/ranging evolution) x `r randomSampling` (uncertainty in brain/diet rate data) x `r length(geographicThresholdVector)` (geographic overlap threshold) x `r length(frugivoryThresholdVector)` (frugivory threshold) x `r length(folivoryThresholdVector)` (folivory threshold) = `r numberSimulations\*randomSampling\*length(geographicThresholdVector)\*length(frugivoryThresholdVector)\*length(folivoryThresholdVector)` sub-models. We stopped computations when the calculation of the likelihood was excessively long (> 1 week). The final sample size thus was of `r min(totModelsWorked[-2])\*10` models.

### Models of species diversification

We investigated how primates diversified over time. Lineage-specific diversification rates were estimated using an updated version of the \*ClaDS\* algorithm [@maliet2019model] boosted for computational speed based on data augmentation techniques [@maliet2020fast]. Particularly, we used \*ClaDS2\*, the model with constant turnover (i.e. constant ratio between extinction and speciation rates). This Bayesian approach considers speciation rate heterogeneity by modeling small shifts in the rate at speciation events. In other words, the daughter lineage is assumed to inherit new speciation rates that is sampled from a log-normal distribution with an expected mean value $log(\alpha \lambda)$ (where $\lambda$ represents the parental speciation rate and $\alpha$ is a trend parameter), and a standard deviation $\sigma$. Three independent chains were run until their respective convergence was validated by a Gelman-Rubin diagnostic criterion [@gelman1992inference]. The analysis relied on the use of a consensus tree of primate phylogeny from @dos2018using. This latter provides a robust phylogenetic tree for 367 primate species (while the 10kTrees primate phylogeny has only `r length(phylo\_init$tip.label)` species).

| Such analysis necessarily depends on a prior estimation of the sample representativeness, that is, the fraction of sampled taxa (present in the phylogenetic tree) among all possible existing ones. @estrada2017impending estimated that, given current knowledge, the primate clade should be composed of 504 species. This means that the current sampling fraction is around 73%. We thus parameterized the \*ClaDS\* algorithm with this value for the estimate sampling fraction. Yet, given that the extant number of primate species is subject to controversy, and because the estimated sampling fraction may affect diversification rate estimations, we replicated our analyses with a range of sampling fractions from 95% down to 60%. At the end of each run, we extracted the maximum of the \*a posteriori\* net diversification rate of each extant primate species, as well as the mean diversification rate (given all lineages) through time.

### Phylogenetic regressions

\hfill

(a) Determining the direction of the selection gradient shaped by interspecific competition

\hfill

To determine the nature of the relationship between species sympatry and relative sizes of brain regions, we fitted Gaussian Pagel's lambda phylogenetic regressions (i.e. a derivative of the Brownian Motion model, for which the phylogenetic variance-covariance matrix has all coefficients but its diagonal multiplied by lambda) for each brain region individually and for frugivorous species only. We used the Pagel's lambda model so as to relax the hypothesis of Brownian Motion since we it included brain areas for which the evolutionary history was best described by competitive models. Here specifically, we considered the least stringent frugivory assessment, with frugivory threshold fixed to `r frugivoryThresholdVector[1]`%, folivory threshold fixed to `r folivoryThresholdVector[1]`%. If, due to data variability, a species did not robustly fit into the categorical classification “frugivorous versus folivorous” (i.e. could be either of the two), it was considered as frugivorous nonetheless.

| The response variable was the relative size of brain areas shown as better described by competitive phylogenetic scenario (see above). Due to data variability, we took the mean of the possible values given the different datasets, and assessed the sensitivity using non-averaged values (see [Phylogenetic regressions: results, stability and assumption]). In this model, the covariates (i.e. continuous predictors) were the average percent of the range surface overlapping with other sympatric frugivorous species, and the number of frugivorous sympatric species (the second was square rooted, to reach symmetrical distribution). For a given species A, sympatry with another species B was considered when species B range overlapped on more than 10% of the range of species A. This was done to reduce noise induced by coarse identification of species range.

\hfill

(b) Diversification and brain size

\hfill

In the same way than explained above, we fitted Gaussian Pagel's lambda phylogenetic regressions of the different relative brain sizes against the diversification rate (i.e. accounting for both, speciation and extinction) estimated for each species by the \*ClaDS\* algorithm. Again, we took the mean of the brain trait values for the main model and assessed the sensitivity by re-running the model several times using non-averaged values in this case.

\hfill

(c) Diversification and sympatry

To determine whether sympatry was associated to specific diversification pattern (and thus if diversification rates were regionalized), we fitted Gaussian Pagel's lambda phylogenetic regressions with the diversification rate as output variable, and used the two metrics for describing sympatry (the average percent of the range surface overlapping with other sympatric frugivorous species, and the number of frugivorous sympatric species) as tested variables, as described in (a).

\hfill

(d) Model implementation

\hfill

(i) Direction of the selection gradient shaped by interspecific competition

\hfill

Models were fitted using the “phylolm” function from the \*phylolm\* package [@phylolm], with the lambda parameter (i.e. indicating whether the trait is subject to selection, or corresponds to Brownian Motion, if $\lambda$ tends towards 1) estimated by maximum-likelihood (argument “model” set to “lambda”). Bootstrapping over `r repetitionBootstrap` independent replicates was done so as to obtain confidence intervals. Other function parameters were set to default. Prior fitting, covariates were symmetrized if necessary. Necessary assumptions on the normal distribution of residuals and homoscedasticity were visually assessed and pointed out no violation (see Appendix [Model assumptions]). We did not observe correlation issue among predictors either (VIF$\_{max}$ < 2, @mundry2014statistical).

\hfill

(ii) Diversification and brain size

\hfill

We could not compute phylogenetic regressions to link diversification and brain traits in frugivorous primates using a frequentist approach because it led to violation of homoscedasticity. Instead, we fitted Bayesian phylogenetic regressions using the "MCMCglmm" function of the \*MCMCglmm\* package [@MCMCglmm]. Each chain was based on a burnin period of `r burnin` iterations, among a total of `r nitt` iterations, and was sampled every `r thin` iterations. We used the least informative priors. Fixed priors were let to default (Gaussian distribution of mean 0 and variance $10^{8}$). Prior on random effects and residuals were set to follow an inverse-Wishart distribution with a variance at limit ($V$) of 1, and a degree of belief ($nu$) of 0.02. We checked model convergence by fitting three chains, and calculated the Gelman-Rubin criterion (max value < `r round(max(gelmanRubinValues), digit=2)`; @gelman1992inference), as well as checked autocorrelation (max absolute value < `r round(max(valueAutoCorr), digit=2)`) using the respective "gelman.diag" and "autocorr.diag" functions from the \*coda\* package [@coda]. In Appendix [Model assumptions], we present trace and distributions of posterior estimates. We further checked the quality of the posterior by visually assessing the Q-Q plot of the posterior with that of a Gaussian distribution of mean 0 and sd 1 (see Appendix [Model assumptions]). We present the estimate together with the 95% credibility interval centered on the mode (Highest Density Posterior, HDP), together with a MCMC p-value (pMCMC) that corresponds to the probability that the estimate ($\beta$) is positive if the mean estimate ($\hat{\beta}$) is negative (i.e. $P(\beta>0|\hat{\beta}<0)$), or if the mean estimate is positive, the probability that the estimate is negative (i.e. $P(\beta<0|\hat{\beta}>0)$).

\hfill

(iii) Diversification and sympatry

We fitted phylogenetic regression as explained in (i). In particular, verification of model assumption and stability pointed out no source of worry (see [Phylogenetic regressions: results, stability and assumption]).

\hfill

(d) Model robustness

\hfill

To assess frequentist model stability with regards to singular points, we computed the DfBetas (variation in estimates) by discarding one observation at a time of the "standard" dataset used to fit the main model, based on the consensus tree.

| To assess the sensitivity to (i) the variability in data and (ii) phylogeny uncertainty, we refitted the models using `r repetitionTrees` phylogenetic trees among the 10000 possible trees from the 10kTrees project. For each of these trees, we fitted the model `r repetitionModels` times, allowing random sampling for data when we had multiple values (e.g. if body mass was provided by different datasets etc.). For the diversification analysis specifically, we also assessed the sensitivity to changes in primate sampling fraction by refitting the models for values ranging between 60 to 95% (as specified before) using the "standard" dataset and the consensus tree.

| The results of these assessment (min-max of estimates) are shown in Appendix [Model stability]. It emphasizes weak sensitivity of the results.

# Acknowledgements

We considerably value the help provided by Jonathan Drury in making some scripts available, but mostly for helping us in solving issues encountered with the use of functions of his own in the \*RPANDA\* package in \*R\*, and that of Marie-Claude Quidoz for assistance for using the CEFE cluster. We thank Simon Benhamou and Manon Clairbaux for discussion and advices on spatial projections, and M. Quéroué, V. Lauret, A. Caizergues and C. Teplistky for feedback on Bayesian computations too. Finally, this work could not have been possible without prior data collection from the IUCN Red List (primate ranging), the 10kTrees project (phylogenetic trees), and Alexandra R. DeCasien and collaborators, Lauren E. Powell and collaborators, Orlin S. Todorov and collaborators, Erik P. Willems and collaborators, Fiona Pearce and collaborators, Navarrete and collaborators, and Cyril C. Grueter who provided primate trait data we used as (supplementary) material with their articles, as well as Nicholas J. Matzke for available algorithm scripts allowing us to implement and better understand the methods. Their indirect input is therefore tremendous. Both authors were supported by a doctoral grant from the \*École Normale Supérieure\*, Paris.

# Authors' contribution

BR conceived the study, collected, cleaned and analyzed the data, drew the figures and wrote the first version of the manuscript and subsequently revised it. BP-L implemented the ClaDS algorithm for our data, helped with running other analyses, and revised the manuscript multiple times. The authors declare having no conflict of interest. All authors gave final approval for publication and agree to be held accountable for the work performed therein.

```{r tableRegGradient, include=TRUE}

results.df\_gradient <- results.df\_gradient[-(6:10),]

rownames(results.df\_gradient) <- NULL

knitr::kable(results.df\_gradient, escape=TRUE, booktabs = TRUE, caption = "Model estimates and significance of phylogenetic regressions to assess the selection gradient direction | Est.=Estimate, CI2.5\\%=Lower border of the CI95\\%, CI97.5\\%=Upper border of the CI95\\%, Sd=Standard deviation, t=Statistics t-value. The brain area (as well as the associated sample size) are indicated prior each list of estimates. The transformation applied to variables are indicated between brackets (logarithm, log, or square-root, sqrt), as well as the ponderation by bodymass (/bodymass).") %>%

kableExtra::column\_spec(2:ncol(results.df\_gradient), bold = toPlotBold[-(6:10)]) %>% #Remove brain raw

kableExtra::kable\_styling(latex\_options = "striped") %>%

kableExtra::kable\_styling(latex\_options="scale\_down") #%>%

#kableExtra::kable\_styling(latex\_options = "HOLD\_position")

```

```{r}

#Round numbers table

results.df\_diversification[seq(from=4, to=nrow(results.df\_diversification), by=4),c(2,3,4)] <-apply(results.df\_diversification[seq(from=4, to=nrow(results.df\_diversification), by=4),c(2,3,4)], 2, function(x){roundIntelligent(as.numcharac(x), digit=2)})

results.df\_diversification[seq(from=1, to=nrow(results.df\_diversification), by=4),1] <- gsub("\\(log\\) ", "", results.df\_diversification[seq(from=1, to=nrow(results.df\_diversification), by=4),1])

results.df\_diversification[seq(from=1, to=nrow(results.df\_diversification), by=4),1] <- gsub("\\(\\/bodymass, log\\) ", "", results.df\_diversification[seq(from=1, to=nrow(results.df\_diversification), by=4),1])

```

\newpage

```{r tableRegDiversification, include=TRUE}

results.df\_diversification <- results.df\_diversification[-(5:8),]

rownames(results.df\_diversification) <- NULL

knitr::kable(results.df\_diversification, escape=TRUE, booktabs = TRUE, caption = "Model estimates and significance of Bayesian phylogenetic regressions to assess the correlation between diversification and brain size | Est.=Estimate, HDP2.5\\%=Lower border of the 95\\% Highest Posterior Density, HDP97.5\\%=Upper border of the 95\\% Highest Posterior Density, Eff. samp.=Effective sample (adjusted for autocorrelation). The brain area (as well as the associated sample size) are indicated prior each list of estimates. The logarithm transformation was applied to variable and is indicated between brackets (log), as well as the ponderation by bodymass (/bodymass).") %>%

kableExtra::column\_spec(2:ncol(results.df\_diversification), bold = toPlotBoldDiversification[-(5:8)]) %>% #Remove brain raw

kableExtra::kable\_styling(latex\_options = "striped") %>%

kableExtra::kable\_styling(latex\_options="scale\_down") #%>%

#kableExtra::kable\_styling(latex\_options = "HOLD\_position")

```

\newpage

```{r tableRegDiversificationAndSympatry, include=TRUE}

knitr::kable(results.df\_diversificationAndSympatry, escape=TRUE, booktabs = TRUE,

caption = "Model estimates and significance of phylogenetic regressions to assess the correlation between diversification and sympatry | Est.=Estimate, CI2.5%=Lower border of the CI95%, CI97.5%=Upper border of the CI95%, Sd= Standard deviation, t= Statitsitics t-vale. The brain area (as well as the associated sample size)

are indicate prior each list of estimates. the transformation (logarithm or square-root) if indicated in parenthese by the abbreviation (log or sqrt).") %>%

kableExtra::column\_spec(2:ncol(results.df\_diversificationAndSympatry), bold = toPlotBoldSympatry) %>%

kableExtra::kable\_styling(latex\_options = "striped") %>%

kableExtra::kable\_styling(latex\_options="scale\_down")

```

\clearpage

```{r prepareMap}

#Reimport areas with cropping

centroid <- matrix(NA, ncol=2, nrow=length(areaName))

for(i in 1:length(areaName)){

areaTransitory <- readOGR(dsn=paste("T:/IUCN\_data\_primate/Geographic\_areas/Shapefiles/",areaName[i],".shp",sep=""))

areaTransitory = clgeo\_Clean(areaTransitory)

areaTransitory <- spTransform(areaTransitory, CRS("+proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0"))

#Have mercator for intersection

areaTransitory <- spTransform(areaTransitory, CRS("+proj=merc +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0"))

areaTransitory <- gIntersection(areaTransitory, worldMap\_mercator, byid=FALSE)

areaTransitory = clgeo\_Clean(areaTransitory)

#Reunite polygon in case

areaTransitory <- gBuffer(areaTransitory, byid=F, width=0)

#back transform to long/lat

areaTransitory <- spTransform(areaTransitory, CRS("+proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0"))

assign(paste("area", i, sep="\_"), areaTransitory)

if(i==1){

centroid[i,] <- c(summary(areaTransitory)$bbox[1,2] + 5, summary(areaTransitory)$bbox[2,1])

}

else if (i==2){

centroid[i,] <- c(summary(areaTransitory)$bbox[1,2] - 5, summary(areaTransitory)$bbox[2,1] - 5)

}

else{

centroid[i,] <- geosphere::centroid(areaTransitory)

}

}

warnings()

```

```{r figmap, include=TRUE, position='H', warning = FALSE, message = FALSE, results= 'hide', fig.width=7, fig.height=7, fig.cap=paste("Geographic areas used for ancestral range reconstruction represented on the Natural Earth projection of the world | Areas were defined as a combination of geographic and environmental criteria relatively to the primate taxonomy following results from [@kamilar2009environmental]: (1) East Madagascar (2) West Madagascar (3) West Africa (4) Central Africa (5) East/South Africa (6) Central America (7) North South-America (8) South South-America (9) West Asia (10) Central/East Asia (11) South Asia (12) Asian peninsula and islands. Note that the north part of Africa and the south of Europe were discarded despite the presence of one primate species (\*Macaca sylvanus\*), because of its geographical complete isolation and repeated intervention of human people in population maintenance [@modolo2005phylogeography]. Hence, \*Macaca Sylvanus\* is not considered in this study.", sep="")}

#abbreviation name are available in https://proj.org/operations/projections/index.html

#help for graticule: https://rpsychologist.com/working-with-shapefiles-projections-and-world-maps-in-ggplot

proj.map <- "+proj=natearth +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0"

centroid <- as.data.frame(centroid)

colnames(centroid) <- c("long", "lat")

centroid$nudge\_y <- 0

centroid$nudge\_y[5] <- -5

library(tidyverse)

library(ggspatial)

library(rnaturalearth)

world <- ne\_countries(scale = "medium", returnclass = "sf")

# graticule (Robin)

grat <- readOGR(dsn="C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Formap", layer="ne\_110m\_graticules\_15")

grat\_df <- fortify(grat)

bbox <- readOGR(dsn="C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Formap", layer="ne\_110m\_wgs84\_bounding\_box")

bbox\_df<- fortify(bbox)

bbox.sf <- sfheaders::sf\_polygon( obj = bbox\_df

, x = "long"

, y = "lat"

, polygon\_id = "group"

)

sf::st\_crs(bbox.sf) <- st\_crs(world)

grat\_df <- grat\_df[,c(1,2,6)]

colnames(grat\_df) <- c("x", "y", "linestring\_id")

grat\_df.sf <- sfheaders::sf\_line(obj = grat\_df)

sf::st\_crs(grat\_df.sf) <- st\_crs(world)

#Main map

map <- ggplot() +

geom\_sf(data = bbox.sf, fill=adjustcolor("white", alpha.f=0.45), col="white") +

geom\_sf(data = grat\_df.sf, col="darkgrey", lty=2) +

geom\_sf(data = world, fill = "black", col = "white") +

theme(panel.background=element\_rect(fill="white"),

panel.grid = element\_blank(),

axis.title.x=element\_blank(),

axis.text.x=element\_blank(),

axis.ticks.x=element\_blank(),

axis.title.y=element\_blank(),

axis.text.y=element\_blank(),

axis.ticks.y=element\_blank()

)

#Plot polygons areas

mergedArea <- c()

areaNameCorrected <- c(

"East Madagascar",

"West Madagascar",

"West Africa",

"Central Africa",

"East/South Africa",

"Central America",

"Northern South America",

"Southern South America",

"West Asia",

"Central/East Asia",

"South Asia",

"Asian islands")

for(i in 1:length(areaName)){

toFortify <- get(paste("area", i, sep="\_"))

toFortify <- fortify(toFortify)

toFortify$group <-paste(areaName[i], toFortify$piece)

toFortify$colour <- gsub("\_", " ", areaName[i],)#areaNameCorrected[i]#colourArea[i]

mergedArea <- rbind(mergedArea, toFortify)

#add to ggplot

#map <- map + geom\_polygon(data = toFortify, aes(x = long, y = lat, group = group), col = colourArea[i], fill = colourArea[i])

}

centroid.sf <- st\_as\_sf(centroid, coords=c("long", "lat"), crs=st\_crs(world))

library(sfheaders)

mergedArea.sf <- sfheaders::sf\_polygon( obj = mergedArea

, x = "long"

, y = "lat"

, polygon\_id = "group"

)

sf::st\_crs(mergedArea.sf) <- st\_crs(world)

#Readd colour

mergedArea.sf$colour <- unique(mergedArea[,c(7,8)])[,2]

# mergedArea, coords=c("long", "lat"), crs=st\_crs(world))

# st\_cast(mergedArea.sf, to="POLYGON")

map <- map +

geom\_sf(data = mergedArea.sf, aes(fill = colour)) +

geom\_sf\_label(data = centroid.sf, aes(label = 1:nrow(centroid)),

fill = c(colourArea[11], colourArea[12], rep("white", times=10)),

col = c("black", "white", rep("black", times=10)),#colourArea[3:12]),

nudge\_y=centroid$nudge\_y,

label.r = unit(0.4, "lines"),

size = 2) +

coord\_sf(crs = proj.map, expand = F) +

scale\_fill\_manual(values = colourArea, labels=c("Central Africa", "South/East Africa", "West Africa", "Central America", "North of South America", "South of South America", "Central/East Asia", "Asian islands", "South Asia", "West Asia", "East Madagascar", "West Madagascar")) +

labs(fill = "") +

theme(

text = element\_text(size = 10),

legend.key = element\_rect(size = 4),

legend.key.size = unit(1, 'lines'),

legend.position="bottom")

map

# # #Just to see the difference as pinpointed by the warning

# # # geosphere::areaPolygon(areaTransitory)

# # # gArea(areaTransitory)

# # # really false

# # # geosphere::centroid(areaTransitory)

# # # gCentroid(areaTransitory)

# # #ok

# #

# # layout(mat=t(c(1,2)), widths=c(40,40), heights=c(40))

# # par(mar=c(0, 0, 0, 0), mgp=c(2, 0.5, 0), xpd=TRUE)

# #

# # Add figure interaction here if needed

# ####

# ## Fig map

# ####

#

# #Create the map of the geographic area

# #Have background

# maps::map("world", fill=TRUE, col="lightgray", bg="white", border=NA, mar = c(0, 0, 0, 0))#, ylim=c(-60, 50))

#

# # addLabel(x=0.05, y=0.075, label="A", radius=7, circle=TRUE, circle.bg="black", font.col="white")

# #

#

# for(i in 1:length(areaName)){

# plot(get(paste("area", i, sep="\_")), col=colourArea[i], border=colourArea[i], add=TRUE) #border="black",

# }

# #Have borders

# #plot(worldMap, col=NA, border="white",bg="white", lwd=0.1, add=TRUE)

# # for(i in 1:length(areaName)){

# # plot(get(paste("area", i, sep="\_")), col=NA, border="black", add=TRUE)

# # }

# points(x=centroid[,1], y=centroid[,2], pch=19, col=c(colourArea[1], colourArea[2], rep("white", times=10)), cex=1.3)

# points(x=centroid[,1], y=centroid[,2], cex=1.3, col=c("white", "white", rep("black", times=10)))

# text(x=centroid[,1], y=centroid[,2], labels=1:length(areaName), cex=0.5, col=c("white", "white", rep("black", times=10)), adj=c(0.5,0.5))

```

\clearpage

```{r phylogeny, fig.pos='H', include=TRUE, warning = FALSE, message = FALSE, fig.width=7, fig.height=7, fig.cap=paste("Current frugivorous primate sympatry pattern and phylogeny | Primate phylogeny from a consensus tree of 1000 possible trees from the 10kTrees project is depicted in the centre, together with abbreviated species name. The corresponding non-abbreviated names can be found using Appendix Figure \\@ref(fig:figdata). Co-occurring frugivorous (based on a frugivory threshold of ", frugivoryThresholdVector[1], "% and folivory of ", folivoryThresholdVector[1], "%) species are linked by lightgray lines. The geographic area occupied by a species is depicted by the coloured rectangles. Presence was assed given an overlap between the species range and the geographic area of ", geographicThresholdVector[1]\*100, "%.", sep="")}

# ####

# ## Fig interaction

# ####

#Tree

tree <- read.tree("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Raw\_data/Tree/Tree\_diet.nex")

tree <- drop.tip(tree,

tree $tip.label[

which(tree$tip.label

%nin% summaryDataForPlot$SpeciesForPhylogeny)])

#Create geography df

geoBinary <- as.data.frame(summaryDataForPlot[,c(3,which(colnames(summaryDataForPlot)=="geographicCode"))])

colnames(geoBinary) <- c("SpeciesPhylo", "Loc")

#Create species ID

hc = as.hclust(tree)#bird.orders)

labels = hc$labels # name of birds

labels.tordc <- as.data.frame(labels)

colnames(labels.tordc) <- "Name"

labels.tordc <- separate(labels.tordc, col="Name", into=c("Name1", "Name2", "Name3"), sep="\_")

labels.rdc <- apply(labels.tordc, 1, function(x){

if(!is.na(x[3])){

paste(toupper(substr(x[1], 1, 1)), ". ", substr(x[2], 1, 3), ". ", substr(x[3], 1, 1), ".", sep="")

} else{paste(toupper(substr(x[1], 1, 1)), ". ", substr(x[2], 1, 3), sep="")

}

}

)

#Match to have right order for geography

locationSpecies <- geoBinary$Loc[match(labels, geoBinary$SpeciesPhylo)]

colLoc <- colourArea

#Match to have diet

dietSpecies <- summaryDataForPlot$DietaryGuild[match(labels, summaryDataForPlot$SpeciesForPhylogeny)]

#Getting species labels abbreviated

speciesLabels <- hc$labels#Should be in the tree order

#Create the circos plot linking species based on their diet and geography

circos.clear()

circos.par(gap.degree=0, gap.after=0, cell.padding=c(0,0,0,0), track.margin = c(0, 0), "canvas.xlim" = c(-1.1, 1.1), "canvas.ylim" = c(-1.1, 1.1))

circos.initialize(speciesLabels, xlim = c(0, 1))

# #Add family

# circos.track(ylim = c(0, 1), bg.border = NA, track.height = 0.1, track.margin=c(0.01, 0.01),

# panel.fun = function(x, y) {

# i=CELL\_META$sector.numeric.index

# circos.rect(CELL\_META$cell.xlim[1],CELL\_META$cell.ylim[1], CELL\_META$cell.xlim[2],CELL\_META$cell.ylim[2], col="white", border=NA)

# })

#

# circos.track(ylim = c(0, 1), bg.border = NA, track.height = 0.1, track.margin=c(0.01, 0.01),

# panel.fun = function(x, y) {

# i=CELL\_META$sector.numeric.index

# circos.rect(CELL\_META$cell.xlim[1],CELL\_META$cell.ylim[1], CELL\_META$cell.xlim[2],CELL\_META$cell.ylim[2], col="white", border=NA)

# })

#

# for(i in 1:length(unique(summaryDataForPlot$Family))){

# speciesForPlotFamily <- speciesLabels[speciesLabels %in% summaryDataForPlot$SpeciesForPhylogeny[summaryDataForPlot$Family %in% unique(summaryDataForPlot$Family)[i]]]

# if(length(speciesForPlotFamily) > 0){

# highlight.sector(speciesForPlotFamily, track.index = 1, col = adjustcolor("black", alpha.f=0.15), padding = c(0, 0.2, 0, 0),

# text = unique(summaryDataForPlot$Family)[i], cex = 0.5, text.col = "white", niceFacing = TRUE)

# }

# }

#Species name + area it belongs to

circos.track(ylim = c(0, 1), bg.border = NA, track.height = 0.1, track.margin=c(0.01, 0.01),

panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

circos.text(CELL\_META$xcenter, 1, labels.rdc[i], adj = c(0, 0),

facing = "clockwise", niceFacing = TRUE,

col = "black", cex = 0.75, font=3)

geo <- as.numcharac(unlist(strsplit(locationSpecies[i], "")))

for(g in 1:length(geo)){

if(geo[g]==1){

#circos.points(CELL\_META$xcenter,0.75/length(geo)\*(2\*g-1)/2, col=as.character(colLoc[g]), pch=19, cex=0.2)

circos.rect(CELL\_META$cell.xlim[1],0.75/length(geo)\*(g-1), CELL\_META$cell.xlim[2], 0.75/length(geo)\*(g), col=as.character(colLoc[g]), border=NA)

}

}

})

#Plot the geographic links

for(i in 1:length(speciesLabels)){

#locI <- which(strsplit(locationSpecies[i], "")==1)

for(j in i:length(speciesLabels)){

#locJ <- which(strsplit(locationSpecies[j], "")==1)

product <- as.numcharac(unlist(strsplit(locationSpecies[j], "")))\*as.numcharac(unlist(strsplit(locationSpecies[i], "")))

if(i==j|(length(unique(product))==1&unique(product)[1]==0)){

#Do nothing

}

else{

if(dietSpecies[i]=="Fruit"&dietSpecies[i]==dietSpecies[j]){

# colour <- as.data.frame(table(colLoc[which(product==1)]))

# if(is.finite(max(colour$Freq))){

# }else{

# print(c(i,j))

# }

# colour <- colour[colour$Freq==max(colour$Freq),1][1]

circos.link(speciesLabels[i], runif(1, 0, 1), speciesLabels[j], runif(1, 0, 1), lwd=0.2, col=adjustcolor("black",alpha.f=0.15))#"lightgray")#adjustcolor(as.character(colour), alpha.f=0.9))

}

else{

#circos.link(speciesLabels[i], runif(1, 0, 1), speciesLabels[j], runif(1, 0, 1), lwd=1, col="lightgray")

}

}

}

}

circos.clear()

#Plot the phylogenetic tree in a new circular plot

n = length(labels) # number of species

dend = as.dendrogram(hc)

par(new = TRUE) # <- magic

circos.par("canvas.xlim" = c(-1.05, 1.05), "canvas.ylim" = c(-1.25, 1.25))

circos.initialize("a", xlim = c(0, n)) # only one sector

# circos.track(ylim = c(0, 1), bg.border = NA, track.height = 0.3,

# panel.fun = function(x, y) {

# for(i in seq\_len(n)) {

# circos.text(i-0.5, 0, labels.rdc[i], adj = c(0, 0.5),

# facing = "clockwise", niceFacing = TRUE,

# col = "black", cex = 0.2, font=3)

# }

# })

#suppressPackageStartupMessages(library(dendextend))

#dend = color\_branches(dend, k = 6, col = 1:6)

dend\_height = attr(dend, "height")

circos.track(ylim = c(0, dend\_height), bg.border = NA,

track.height = 0.95, panel.fun = function(x, y) {

circos.dendrogram(dend)

})

circos.clear()

```

\clearpage

```{r figbrain, fig.pos='H', include=TRUE, warning = FALSE, message = FALSE, fig.width=7, fig.height=7, fig.cap=paste("(Left) EQ or relative brain size value among frugivorous primates (Right) Studied brain areas | (Left) The circular rows are indicated by the colours which match a specific brain area. The darker background emphasises when values are above average, while the lighter background emphasises when values are below average. The mean value (after scaling and based on one random sampling among possible values, but see \\@ref(fig:figvariabilitydata) for visualization of measure variability) for the Encephalization Quotient (EQ) or relative size of brain parts, when available, is depicted by a plain circle for frugivorous species. The frugivorous threshold was fixed to ", frugivoryThresholdVector[1], "% and folivory to ", folivoryThresholdVector[1], "%. (Right) A 3D brain from \*Homo sapiens\* is depicted (\*neurobase\* package [@neurobase], \*misc3d\* package [@misc3d]). The arrows indicate the sagital and frontal axes. Studied brain area are coloured, although the neocortex was not coloured for readability since it corresponds to the external layer of the cerebral hemisphere. In short, the MOB is involved in immediate olfactory information processing, the Neocortex and the Cerebellum support a working memory and memory consolidation processes [@wiltgen2004new; @koziol2014consensus; @sokolov2017cerebellum], and the Hippocampus supports a working memory and a long-term spatio-temporal memory [@burgess2002human]. The Striatum is involved in social information processing [@baez2013role].", sep="")}

library(RColorBrewer)

colourVector <- c("darkgrey", brewer.pal(n = 5, name = "Set1")[1:5])

colourVectorbis <- c("lightgray", brewer.pal(n = 5, name = "Pastel1")[1:5])

colour.circle.points <- c("black", "darkred", "darkblue", "darkgreen", "purple4", "orange4")

###

## Fig brain values / circular

###

summaryDataForPlot$EQ <- summaryDataForPlot$Brain\*1.036\*(10\*\*-3)/(0.085\*summaryDataForPlot$Bodymass\*\*0.775)

summaryDataForPlot$ratioNeocortex <- summaryDataForPlot$Neocortex/summaryDataForPlot$Bodymass

summaryDataForPlot$ratioHippocampus <- summaryDataForPlot$Hippocampus/summaryDataForPlot$Bodymass

summaryDataForPlot$ratioCerebellum <- summaryDataForPlot$Cerebellum/summaryDataForPlot$Bodymass

summaryDataForPlot$ratioStriatum <- summaryDataForPlot$Striatum/summaryDataForPlot$Bodymass

summaryDataForPlot$ratioMOB <- summaryDataForPlot$MOB/summaryDataForPlot$Bodymass

#Brain data

relativeValueEQ <- scale(summaryDataForPlot$EQ[match(speciesLabels, summaryDataForPlot$SpeciesForPhylogeny)])#runif(length(speciesLabels), -1, 1)

relativeValueNeocortex <- scale(summaryDataForPlot$ratioNeocortex[match(speciesLabels, summaryDataForPlot$SpeciesForPhylogeny)])

relativeValueHippocampus <- scale(summaryDataForPlot$ratioHippocampus[match(speciesLabels, summaryDataForPlot$SpeciesForPhylogeny)])

relativeValueCerebellum <- scale(summaryDataForPlot$ratioCerebellum[match(speciesLabels, summaryDataForPlot$SpeciesForPhylogeny)])

relativeValueStriatum <- scale(summaryDataForPlot$ratioStriatum[match(speciesLabels, summaryDataForPlot$SpeciesForPhylogeny)])

relativeValueMOB <- scale(summaryDataForPlot$ratioMOB[match(speciesLabels, summaryDataForPlot$SpeciesForPhylogeny)])

layout(mat=cbind(c(1,1), c(2,3)), widths=c(35,15), heights=c(15,35))

par(mar=c(0, 0, 0, 0), mgp=c(2, 0.5, 0), xpd=TRUE)

# plot(0, 0, type="n")

library(circlize)

circos.clear()

circos.par(gap.degree=0, gap.after=0, cell.padding=c(0,0,0,0), track.margin=c(0, 0))

circos.initialize(speciesLabels, xlim = c(0, 1))

#Species name

circos.track(ylim = c(0, 20), bg.border = NA, track.height = 0.05, track.margin=c(0.01, 0.1),

panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

circos.text(CELL\_META$xcenter, 0, labels.rdc[i], adj = c(0, 0),

facing = "clockwise", niceFacing = TRUE,

col = "black", cex = 0.6, font=3)

})

#

#Background

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVector[1], border=colourVector[1])

}, track.height = 1/15)

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVectorbis[1], border=colourVectorbis[1])

}, track.height = 1/15)

#Background

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVector[2], border=colourVector[2])

}, track.height = 1/15)

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVectorbis[2], border=colourVectorbis[2])

}, track.height = 1/15)

#Background

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVector[3], border=colourVector[3])

}, track.height = 1/15)

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVectorbis[3], border=colourVectorbis[3])

}, track.height = 1/15)

#Background

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVector[4], border=colourVector[4])

}, track.height = 1/15)

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVectorbis[4], border=colourVectorbis[4])

}, track.height = 1/15)

#Background

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVector[5], border=colourVector[5])

}, track.height = 1/15)

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVectorbis[5], border=colourVectorbis[5])

}, track.height = 1/15)

#Background

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVector[6], border=colourVector[6])

}, track.height = 1/15)

circos.track(ylim = c(0, 1), bg.border = NA, panel.fun = function(x, y) {

circos.rect(0, 0, 1, 1, col=colourVectorbis[6], border=colourVectorbis[6])

}, track.height = 1/15)

library(plotrix)

#Main circle

for(i in 1:13){

draw.circle(x=0,y=0,0.91-1/15-(i-1)\*1/15, col=NA, border="white")

}

#increment of 0.5

for(i in 1:26){

draw.circle(x=0,y=0,0.91-1/15-(i-1)\*1/15/2, col=NA, border="white", lty=2)

}

#Value

#EQ

absMax <- max(abs(relativeValueEQ), na.rm=TRUE)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=2, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

#circos.rect(0, 0, 1, 1, col=colourPositive, border=colourPositive)

if(is.na(relativeValueEQ[i])){} else{

if(relativeValueEQ[i] > 0 & dietSpecies[i]=="Fruit"){

circos.points(CELL\_META$xcenter, relativeValueEQ[i]/absMax, pch=19, col=colour.circle.points[1], cex=0.7)

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueEQ[i]/absMax, col=colour.circle.points[1], lty=3)

}

else if(relativeValueEQ[i] > 0 & dietSpecies[i]=="Leaf"){

circos.points(CELL\_META$xcenter, relativeValueEQ[i]/absMax, pch=21, col=colour.circle.points[1], bg="white", cex=0.7)

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueEQ[i]/absMax, col=colour.circle.points[1], lty=3)

}

else{}

}

}, track.height = 0.1)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=3, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

if(is.na(relativeValueEQ[i])){} else{

#circos.rect(0, 0, 1, 1, col=colourNegative, border=colourNegative)

if(relativeValueEQ[i] <= 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueEQ[i]/absMax, col=colour.circle.points[1], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueEQ[i]/absMax, pch=19, col=colour.circle.points[1], cex=0.7)

}

else if(relativeValueEQ[i] <= 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueEQ[i]/absMax, col=colour.circle.points[1], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueEQ[i]/absMax, pch=21, col=colour.circle.points[1], bg="white", cex=0.7)

}

else{}

}

}, track.height = 0.1)

#Striatum

absMax <- max(abs(relativeValueStriatum), na.rm=TRUE)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=4, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

#circos.rect(0, 0, 1, 1, col=colourPositive, border=colourPositive)

if(is.na(relativeValueStriatum[i])){} else{

if(relativeValueStriatum[i] > 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueStriatum[i]/absMax, col=colour.circle.points[2], lty=3)

circos.points(CELL\_META$xcenter, relativeValueStriatum[i]/absMax, pch=19, col=colour.circle.points[2], cex=0.65)

}

else if(relativeValueStriatum[i] > 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueStriatum[i]/absMax, col=colour.circle.points[2], lty=3)

circos.points(CELL\_META$xcenter, relativeValueStriatum[i]/absMax, pch=21, col=colour.circle.points[2], bg="white", cex=0.65)

}

else{}

}

}, track.height = 0.1)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=5, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

if(is.na(relativeValueStriatum[i])){} else{

#circos.rect(0, 0, 1, 1, col=colourNegative, border=colourNegative)

if(relativeValueStriatum[i] <= 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueStriatum[i]/absMax, col=colour.circle.points[2], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueStriatum[i]/absMax, pch=19, col=colour.circle.points[2], cex=0.65)

}

else if(relativeValueStriatum[i] <= 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueStriatum[i]/absMax, col=colour.circle.points[2], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueStriatum[i]/absMax, pch=21, col=colour.circle.points[2], bg="white", cex=0.65)

}

else{}

}

}, track.height = 0.1)

#MOB

absMax <- max(abs(relativeValueMOB), na.rm=TRUE)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=6, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

#circos.rect(0, 0, 1, 1, col=colourPositive, border=colourPositive)

if(is.na(relativeValueMOB[i])){} else{

if(relativeValueMOB[i] > 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueMOB[i]/absMax, col=colour.circle.points[3], lty=3)

circos.points(CELL\_META$xcenter, relativeValueMOB[i]/absMax, pch=19, col=colour.circle.points[3], cex=0.65)

}

else if(relativeValueMOB[i] > 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueMOB[i]/absMax, col=colour.circle.points[3], lty=3)

circos.points(CELL\_META$xcenter, relativeValueMOB[i]/absMax, pch=21, col=colour.circle.points[3], bg="white", cex=0.65)

}

else{}

}

}, track.height = 0.1)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=7, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

if(is.na(relativeValueMOB[i])){} else{

#circos.rect(0, 0, 1, 1, col=colourNegative, border=colourNegative)

if(relativeValueMOB[i] <= 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueMOB[i]/absMax, col=colour.circle.points[3], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueMOB[i]/absMax, pch=19, col=colour.circle.points[3], cex=0.65)

}

else if(relativeValueMOB[i] <= 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueMOB[i]/absMax, col=colour.circle.points[3], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueMOB[i]/absMax, pch=21, col=colour.circle.points[3], bg="white", cex=0.65)

}

else{}

}

}, track.height = 0.1)

#Neocortex

absMax <- max(abs(relativeValueNeocortex), na.rm=TRUE)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=8, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

#circos.rect(0, 0, 1, 1, col=colourPositive, border=colourPositive)

if(is.na(relativeValueNeocortex[i])){} else{

if(relativeValueNeocortex[i] > 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueNeocortex[i]/absMax, col=colour.circle.points[4], lty=3)

circos.points(CELL\_META$xcenter, relativeValueNeocortex[i]/absMax, pch=19, col=colour.circle.points[4], cex=0.6)

}

else if(relativeValueNeocortex[i] > 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueNeocortex[i]/absMax, col=colour.circle.points[4], lty=3)

circos.points(CELL\_META$xcenter, relativeValueNeocortex[i]/absMax, pch=21, col=colour.circle.points[4], bg="white", cex=0.6)

}

else{}

}

}, track.height = 0.1)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=9, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

if(is.na(relativeValueNeocortex[i])){} else{

#circos.rect(0, 0, 1, 1, col=colourNegative, border=colourNegative)

if(relativeValueNeocortex[i] <= 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueNeocortex[i]/absMax, col=colour.circle.points[4], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueNeocortex[i]/absMax, pch=19, col=colour.circle.points[4], cex=0.6)

}

else if(relativeValueNeocortex[i] <= 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueNeocortex[i]/absMax, col=colour.circle.points[4], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueNeocortex[i]/absMax, pch=21, col=colour.circle.points[4], bg="white", cex=0.6)

}

else{}

}

}, track.height = 0.1)

#Hippocampus

absMax <- max(abs(relativeValueHippocampus), na.rm=TRUE)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=10, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

#circos.rect(0, 0, 1, 1, col=colourPositive, border=colourPositive)

if(is.na(relativeValueHippocampus[i])){} else{

if(relativeValueHippocampus[i] > 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueHippocampus[i]/absMax, col=colour.circle.points[5], lty=3)

circos.points(CELL\_META$xcenter, relativeValueHippocampus[i]/absMax, pch=19, col=colour.circle.points[5], cex=0.55)

}

else if(relativeValueHippocampus[i] > 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueHippocampus[i]/absMax, col=colour.circle.points[5], lty=3)

circos.points(CELL\_META$xcenter, relativeValueHippocampus[i]/absMax, pch=21, col=colour.circle.points[5], bg="white", cex=0.55)

}

else{}

}

}, track.height = 0.1)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=11, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

if(is.na(relativeValueHippocampus[i])){} else{

#circos.rect(0, 0, 1, 1, col=colourNegative, border=colourNegative)

if(relativeValueHippocampus[i] <= 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueHippocampus[i]/absMax, col=colour.circle.points[5], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueHippocampus[i]/absMax, pch=19, col=colour.circle.points[5], cex=0.55)

}

else if(relativeValueHippocampus[i] <= 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueHippocampus[i]/absMax, col=colour.circle.points[5], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueHippocampus[i]/absMax, pch=21, col=colour.circle.points[5], bg="white", cex=0.55)

}

else{}

}

}, track.height = 0.1)

#Cerebellum

absMax <- max(abs(relativeValueCerebellum), na.rm=TRUE)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=12, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

#circos.rect(0, 0, 1, 1, col=colourPositive, border=colourPositive)

if(is.na(relativeValueCerebellum[i])){} else{

if(relativeValueCerebellum[i] > 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueCerebellum[i]/absMax, col=colour.circle.points[6], lty=3)

circos.points(CELL\_META$xcenter, relativeValueCerebellum[i]/absMax, pch=19, col=colour.circle.points[6], cex=0.5)

}

else if(relativeValueCerebellum[i] > 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 0, CELL\_META$xcenter, relativeValueCerebellum[i]/absMax, col=colour.circle.points[6], lty=3)

circos.points(CELL\_META$xcenter, relativeValueCerebellum[i]/absMax, pch=21, col=colour.circle.points[6], bg="white", cex=0.5)

}

else{}

}

}, track.height = 0.1)

circos.track(ylim = c(0, 1), bg.border = NA, track.index=13, panel.fun = function(x, y) {

i=CELL\_META$sector.numeric.index

if(is.na(relativeValueCerebellum[i])){} else{

#circos.rect(0, 0, 1, 1, col=colourNegative, border=colourNegative)

if(relativeValueCerebellum[i] <= 0 & dietSpecies[i]=="Fruit"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueCerebellum[i]/absMax, col=colour.circle.points[6], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueCerebellum[i]/absMax, pch=19, col=colour.circle.points[6], cex=0.5)

}

else if(relativeValueCerebellum[i] <= 0 & dietSpecies[i]=="Leaf"){

circos.segments(CELL\_META$xcenter, 1, CELL\_META$xcenter, 1 + relativeValueCerebellum[i]/absMax, col=colour.circle.points[6], lty=3)

circos.points(CELL\_META$xcenter, 1 + relativeValueCerebellum[i]/absMax, pch=21, col=colour.circle.points[6], bg="white", cex=0.5)

}

else{}

}

}, track.height = 0.1)

#Empty plot

par(mar=c(0, 0, 0, 0), mgp=c(2, 0.5, 0), xpd=TRUE)

plot(0, 0, xlim=c(0,1), ylim=c(0,1), xlab="", ylab="", las=1, type="n", tcl=-0.25, bty="n",

xaxt="n",xaxs="i",yaxs="i", yaxt="n", xpd=TRUE)

# #Add brain

library(png)

brainIMG <- readPNG("C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Plots/3dplot.png")

addImg(brainIMG, x = 0.425, y = 0.65, width = 1.3)

#Add circle contour

draw.circle(x=0.45,y=0.55,0.435, col=NA, border="lightgray", lwd=2)

draw.circle(x=0.45,y=0.55,0.405, col=NA, border="lightgray", lwd=2)

par(mar=c(0, 0, 0, 0), mgp=c(2, 0.5, 0), xpd=TRUE)

plot(0, 0, xlim=c(0,1), ylim=c(0,1), xlab="", ylab="", las=1, type="n", tcl=-0.25, bty="n",

xaxt="n",xaxs="i",yaxs="i", yaxt="n", xpd=TRUE)

colourHip=colourVector[5]

colourCereb=colourVector[6]

colourOlf=colourVector[3]

colourStri=colourVector[2]

legend(x=0.1, y=0.8, legend = c("EQ", "Striatum", "MOB", "Hippocampus", "Neocortex", "Cerebellum", "Frugivorous species", "Folivorous species"), cex = 1, fill = c(colourVector, NA, NA), pch=c(NA, NA, NA, NA, NA, NA, 19, 1), border=c("black", "black", "black", "black", "black", "black", "white", "white"), col=c(NA, NA, NA, NA, NA, NA, "black", "black"), bty="n", seg.len=0.75)

```

\clearpage

```{r figresultsevolution, fig.pos='H', include=TRUE, warning = FALSE, message = FALSE, fig.width=7, fig.height=6, fig.cap="Relative support of fitted models of trait evolution for each brain part | Plotted is the AICc weight, a measure of relative support for a given model, for non-competitive (BM, OU, EB) and competitive (MC, DD$\_{lin}$, DD$\_{exp}$) models. The points represent the average AICc weight obtained (when considering the six models from a same run), while the vertical bars indicate the standard deviation given all tested conditions (see [Models of trait evolution: does interspecific interactions shape brain size evolution?])."}

layout(mat=rbind(c(1,2,3), c(4,5,6)), widths=c(5,5,5), heights=c(5,5))

par(mar=c(3.5, 3, 2, 0.5), mgp=c(2, 0.5, 0), xpd=TRUE)

#note: 1= second run for frugivory 20%

#note: \_2= first run for frugivory 20%

## EQ

plot(

x=0, y=0, xlab="", ylab="AICc weight", cex.sub=1.6,

xlim=c(0,7), ylim=c(0,1.2),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=7, xintsmall=0.25, xintbig=1, ymin=0, ymax=1, yintsmall=0.05, yintbig=0.2, axisPlot=FALSE)

axis(side=2, at=seq(from=0, to=1, by=0.2), labels=seq(from=0, to=1, by=0.2), las=2, tcl=-0.25)

text(x=1:6, y=rep(-0.1, times=6), labels=models, cex=0.8, xpd=TRUE)

for(i in 1:6){

meanPt <- mean(as.numcharac(summaryEQFrugivory[, ncol(summaryEQFrugivory)-6+i]))

sd <- sd(as.numcharac(summaryEQFrugivory[, ncol(summaryEQFrugivory)-6+i]))

#sd <- sd/nrow(summaryEQFrugivory) #error not sd

errorBars(location=i, meanPt=meanPt, barValue=sd, refUnit=1, col="black", minValue=0, maxValue=1, horiz=FALSE)

points(x=i, y=meanPt, pch=19, col=colourModels[i], xpd=TRUE)

}

#b and r are the rate for density dependance (DD) of the speciation rate. If positive, positive DD, otherwise, negative.

#add their values below:

text(x=c(4,5,6), y=c(-0.2, -0.2),

labels=c(

"r ~",

round(mean(as.numcharac(summaryEQFrugivory$DDlingeo.b)), digit=3),

round(mean(as.numcharac(summaryEQFrugivory$DDexpgeo.r)), digit=3)

), cex=0.75, xpd=TRUE)

draw.circle(x=0.3,y=1.1,0.25, col=colNum[1], border=NA)

text(x=0.3, y=1.1, labels="1", xpd=TRUE, col="white", font=2)

text(x=3.5, y=1.1, labels="EQ", xpd=TRUE, col="black", font=2, cex=1.5)

##-------------

##------------

#Striatum

plot(

x=0, y=0, xlab="", ylab="", cex.sub=1.6,

xlim=c(0,7), ylim=c(0,1.2),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=7, xintsmall=0.25, xintbig=1, ymin=0, ymax=1, yintsmall=0.05, yintbig=0.2, axisPlot=FALSE)

#axis(side=2, at=seq(from=0, to=1, by=0.2), labels=seq(from=0, to=1, by=0.2), las=2, tcl=-0.25)

segments(x0 = -1, x1 = -1, y0 = 0, y1 = 1, lty = 2, col = colNum[2])

text(x=1:6, y=rep(-0.1, times=6), labels=models, cex=0.8, xpd=TRUE)

for(i in 1:6){

meanPt <- mean(as.numcharac(summaryStriatumFrugivory[, ncol(summaryStriatumFrugivory)-6+i]))

sd <- sd(as.numcharac(summaryStriatumFrugivory[, ncol(summaryStriatumFrugivory)-6+i]))

#sd <- sd/nrow(summaryStriatumFrugivory) #error not sd

errorBars(location=i, meanPt=meanPt, barValue=sd, refUnit=1, col="black", minValue=0, maxValue=1, horiz=FALSE)

points(x=i, y=meanPt, pch=19, col=colourModels[i], xpd=TRUE)

}

draw.circle(x=0.3,y=1.1,0.25, col=colNum[2], border=NA)

text(x=0.3, y=1.1, labels="2", xpd=TRUE, col="white", font=2)

text(x=3.5, y=1.1, labels="Striatum", xpd=TRUE, col="black", font=2, cex=1.5)

#b and r are the rate for density dependance (DD) of the speciation rate. If positive, positive DD, otherwise, negative.

#add their values below:

text(x=c(4,5,6), y=c(-0.2, -0.2),

labels=c(

"r ~",

round(mean(as.numcharac(summaryStriatumFrugivory$DDlingeo.b)), digit=3),

round(mean(as.numcharac(summaryStriatumFrugivory$DDexpgeo.r)), digit=3)

), cex=0.75, xpd=TRUE)

##------------

##-------------

#MOB

plot(

x=0, y=0, xlab="", ylab="", cex.sub=1.6,

xlim=c(0,7), ylim=c(0,1.2),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=7, xintsmall=0.25, xintbig=1, ymin=0, ymax=1, yintsmall=0.05, yintbig=0.2, axisPlot=FALSE)

#axis(side=2, at=seq(from=0, to=1, by=0.2), labels=seq(from=0, to=1, by=0.2), las=2, tcl=-0.25)

segments(x0 = -1, x1 = -1, y0 = 0, y1 = 1, lty = 2, col = colNum[3])

text(x=1:6, y=rep(-0.1, times=6), labels=models, cex=0.8, xpd=TRUE)

for(i in 1:6){

meanPt <- mean(as.numcharac(summaryMOBFrugivory[, ncol(summaryMOBFrugivory)-6+i]))

sd <- sd(as.numcharac(summaryMOBFrugivory[, ncol(summaryMOBFrugivory)-6+i]))

#sd <- sd/nrow(summaryMOBFrugivory) #error not sd

errorBars(location=i, meanPt=meanPt, barValue=sd, refUnit=1, col="black", minValue=0, maxValue=1, horiz=FALSE)

points(x=i, y=meanPt, pch=19, col=colourModels[i], xpd=TRUE)

}

draw.circle(x=0.3,y=1.1,0.25, col=colNum[3], border=NA)

text(x=0.3, y=1.1, labels="3", xpd=TRUE, col="white", font=2)

text(x=3.5, y=1.1, labels="MOB", xpd=TRUE, col="black", font=2, cex=1.5)

#b and r are the rate for density dependance (DD) of the speciation rate. If positive, positive DD, otherwise, negative.

#add their values below:

text(x=c(4,5,6), y=c(-0.2, -0.2),

labels=c(

"r ~",

round(mean(as.numcharac(summaryMOBFrugivory$DDlingeo.b)), digit=3),

round(mean(as.numcharac(summaryMOBFrugivory$DDexpgeo.r)), digit=3)

), cex=0.75, xpd=TRUE)

##------------

##------------

#Hippocampus

plot(

x=0, y=0, xlab="", ylab="AICc weight", cex.sub=1.6,

xlim=c(0,7), ylim=c(0,1.2),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=7, xintsmall=0.25, xintbig=1, ymin=0, ymax=1, yintsmall=0.05, yintbig=0.2, axisPlot=FALSE)

axis(side=2, at=seq(from=0, to=1, by=0.2), labels=seq(from=0, to=1, by=0.2), las=2, tcl=-0.25)

text(x=1:6, y=rep(-0.1, times=6), labels=models, cex=0.8, xpd=TRUE)

for(i in 1:6){

meanPt <- mean(as.numcharac(summaryHippocampusFrugivory[, ncol(summaryHippocampusFrugivory)-6+i]))

sd <- sd(as.numcharac(summaryHippocampusFrugivory[, ncol(summaryHippocampusFrugivory)-6+i]))

#sd <- sd/nrow(summaryHippocampusFrugivory) #error not sd

errorBars(location=i, meanPt=meanPt, barValue=sd, refUnit=1, col="black", minValue=0, maxValue=1, horiz=FALSE)

points(x=i, y=meanPt, pch=19, col=colourModels[i], xpd=TRUE)

}

draw.circle(x=0.3,y=1.1,0.25, col=colNum[4], border=NA)

text(x=0.3, y=1.1, labels="6", xpd=TRUE, col="white", font=2)

text(x=3.5, y=1.1, labels="Hippocampus", xpd=TRUE, col="black", font=2, cex=1.5)

#b and r are the rate for density dependance (DD) of the speciation rate. If positive, positive DD, otherwise, negative.

#add their values below:

text(x=c(4,5,6), y=c(-0.2, -0.2),

labels=c(

"r ~",

round(mean(as.numcharac(summaryHippocampusFrugivory$DDlingeo.b)), digit=3),

round(mean(as.numcharac(summaryHippocampusFrugivory$DDexpgeo.r)), digit=3)

), cex=0.75, xpd=TRUE)

##------------

##-------------

#Neocortex

plot(

x=0, y=0, xlab="", ylab="", cex.sub=1.6,

xlim=c(0,7), ylim=c(0,1.2),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=7, xintsmall=0.25, xintbig=1, ymin=0, ymax=1, yintsmall=0.05, yintbig=0.2, axisPlot=FALSE)

#axis(side=2, at=seq(from=0, to=1, by=0.2), labels=seq(from=0, to=1, by=0.2), las=2, tcl=-0.25)

segments(x0 = -1, x1 = -1, y0 = 0, y1 = 1, lty = 2, col = colNum[5])

text(x=1:6, y=rep(-0.1, times=6), labels=models, cex=0.8, xpd=TRUE)

for(i in 1:6){

meanPt <- mean(as.numcharac(summaryNeocortexFrugivory[, ncol(summaryNeocortexFrugivory)-6+i]))

sd <- sd(as.numcharac(summaryNeocortexFrugivory[, ncol(summaryNeocortexFrugivory)-6+i]))

#sd <- sd/nrow(summaryNeocortexFrugivory) #error not sd

errorBars(location=i, meanPt=meanPt, barValue=sd, refUnit=1, col="black", minValue=0, maxValue=1, horiz=FALSE)

points(x=i, y=meanPt, pch=19, col=colourModels[i], xpd=TRUE)

}

draw.circle(x=0.3,y=1.1,0.25, col=colNum[5], border=NA)

text(x=0.3, y=1.1, labels="5", xpd=TRUE, col="white", font=2)

text(x=3.5, y=1.1, labels="Neocortex", xpd=TRUE, col="black", font=2, cex=1.5)

#b and r are the rate for density dependance (DD) of the speciation rate. If positive, positive DD, otherwise, negative.

#add their values below:

text(x=c(4,5,6), y=c(-0.2, -0.2),

labels=c(

"r ~",

round(mean(as.numcharac(summaryNeocortexFrugivory$DDlingeo.b)), digit=3),

round(mean(as.numcharac(summaryNeocortexFrugivory$DDexpgeo.r)), digit=3)

), cex=0.75, xpd=TRUE)

##-------------

##-------------

#Cerebellum

plot(

x=0, y=0, xlab="", ylab="", cex.sub=1.6,

xlim=c(0,7), ylim=c(0,1.2),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=7, xintsmall=0.25, xintbig=1, ymin=0, ymax=1, yintsmall=0.05, yintbig=0.2, axisPlot=FALSE)

#axis(side=2, at=seq(from=0, to=1, by=0.2), labels=seq(from=0, to=1, by=0.2), las=2, tcl=-0.25)

segments(x0 = -1, x1 = -1, y0 = 0, y1 = 1, lty = 2, col = colNum[6])

text(x=1:6, y=rep(-0.1, times=6), labels=models, cex=0.8, xpd=TRUE)

for(i in 1:6){

meanPt <- mean(as.numcharac(summaryCerebellumFrugivory[, ncol(summaryCerebellumFrugivory)-6+i]))

sd <- sd(as.numcharac(summaryCerebellumFrugivory[, ncol(summaryCerebellumFrugivory)-6+i]))

#sd <- sd/nrow(summaryCerebellumFrugivory) #error not sd

errorBars(location=i, meanPt=meanPt, barValue=sd, refUnit=1, col="black", minValue=0, maxValue=1, horiz=FALSE)

points(x=i, y=meanPt, pch=19, col=colourModels[i], xpd=TRUE)

}

draw.circle(x=0.3,y=1.1,0.25, col=colNum[6], border=NA)

text(x=0.3, y=1.1, labels="7", xpd=TRUE, col="white", font=2)

text(x=3.5, y=1.1, labels="Cerebellum", xpd=TRUE, col="black", font=2, cex=1.5)

#b and r are the rate for density dependance (DD) of the speciation rate. If positive, positive DD, otherwise, negative.

#add their values below:

text(x=c(4,5,6), y=c(-0.2, -0.2),

labels=c(

"r ~",

round(mean(as.numcharac(summaryCerebellumFrugivory$DDlingeo.b)), digit=3),

round(mean(as.numcharac(summaryCerebellumFrugivory$DDexpgeo.r)), digit=3)

), cex=0.75, xpd=TRUE)

###----------------------

```

\newpage

\beginsupplement

\clearpage

# Appendix

## Data availability

Availability of trait and distribution range for the `r length(phylo\_init$tip.label)` primate species represented in the primate phylogeny of the 10kTrees project is depicted in Appendix Figure \@ref(fig:figdata).

## Data variability

We present below the results of the assessments of data variability depending on the considered thresholds (for frugivory, folivory or overlap) and the data set that is used, specifically related to distribution ranges, or anatomical/behavioural traits.

### Sensitivity to variation in distribution range

```{r figcomparison, fig.pos='H', include=TRUE, warning = FALSE, message = FALSE, fig.width=4, fig.height=4, fig.cap=paste("Percent of species with differently identified biogeographic areas in function of the overlap threshold (reference is an overlap threshold of ", geographicThresholdVector[2]/2\*100,"%) | For a given species, a biogeographic area difference means that at least one biogeographic area considers absence/presence of the species while this was not the case with the ", geographicThresholdVector[2]/2\*100, "% threshold. ", geographicThresholdVector[2]/2\*100, "% was chosen as the reference since halfway to the chosen maximum of ", geographicThresholdVector[2]\*100, "%. ", geographicThresholdVector[2]\*100, "% was chosen as the maximum because based on current observations, a species occupied at best three different biogeographic areas.", sep="")}

plot(

x=0, y=0, xlab="Overlap threshold", ylab="Variation percent (relative to 15%)",

xlim=c(thresholdPresenceRange[1],thresholdPresenceRange[length(thresholdPresenceRange)]), ylim=c(0,0.4),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=5/100, xmax=30/100, xintsmall=2.5/200, xintbig=5/100, ymin=0, ymax=0.4, yintsmall=0.01, yintbig=0.1, axisPlot=FALSE)

axis(side=2, at=seq(from=0, to=1, by=0.2), labels=seq(from=0, to=1, by=0.2), las=1, tcl=-0.25)

axis(side=1, at=thresholdPresenceRange, labels=thresholdPresenceRange, las=1, tcl=-0.25)

points(x=thresholdPresenceRange, y=howManyDifferent, pch=19, xpd=TRUE)

lines(x=thresholdPresenceRange, y=howManyDifferent)

```

\clearpage

### Sensitivity to variation in trait value

```{r figvariabilitydata, fig.pos='H', include=TRUE, warning = FALSE, message = FALSE, fig.width=10, fig.height=4, fig.cap="Supplementary Figure. Variation in trait values among reference datasets | Colours are associated to a specific trait: Brain, Hippocampus, Neocortex and Cerebellum refers to the volume of the area (in mm$^{3}$), Body refers to the body mass (in g), Frug. indicates the frugivory rate and Fol. indicates the folivory rate. (A) Correlation: The points depict the coefficient of correlation while the bar depicts the 95% confidence interval (CI). (B) Variability: The points depict the average of the mean ratio $m$ of the absolute of differences with paired values; If we reduce the equation, we have $m=|(v\_{1}^{2}-v\_{2}^{2})|/(2v\_{1}v\_{2})$, where $v\_{1}$ and $v\_{2}$ are the two paired values from two different datasets and are different from 0. If $v\_{1}$ and $v\_{2}$ equal 0, then $m=0$. If $v\_{1}$ or $v\_{2}$ equals 0 (case for the diet rates constrained between [0,1]), then we fixed the null value to 0.01. The bar depicts the standard error. (C) Repeatability: Repeatability was assessed for traits that were included in at least three datasets. Prior calculation, traits were pondered \*within\* species by the \*within\* species max value. The point represents the mean repeatability $r$ calculated as $\\sigma^{2}\_{between}/(\\sigma^{2}\_{between}+\\sigma^{2}\_{within})$, with the $\\sigma^{2}\_{between}$ and $\\sigma^{2}\_{within}$ corresponding the variance \*between\* or \*within\* species. The bar depicts the standard error. For all graphics, sample sizes are indicated above the upper value of the CI/error interval."}

layout(mat=t(c(1,2,3)), widths=c(5,5,3), heights=c(5))

par(mar=c(4, 3, 2, 1), mgp=c(2, 0.5, 0), xpd=TRUE)

cexText <- c(

rep(0.8, times=10),

rep(0.4, times=3),

rep(0.8, times=6),

rep(0.4, times=3),

rep(1, times=1),

rep(0.8, times=6),

rep(1, times=1),

rep(1, times=1)

)

addToY <- c(

rep(0, times=10),

rep(0, times=3),

rep(0, times=6),

rep(0, times=3),

rep(0, times=1),

rep(0, times=6),

rep(-0.05, times=1),

rep(0, times=1)

)

colourWhatCompared <- c("gray", brewer.pal(n = 9, name = "Pastel1")[c(1:5)], "black", brewer.pal(n = 9, name = "Pastel1")[7])#c(brewer.pal(n = length(unique(whatCompared)) - 1, name = "Pastel1"), "darkgrey")

###------

### CORRELATION

#Vectors to save results

barLower <- rep(NA, times=length(colNumTest))

barUpper <- rep(NA, times=length(colNumTest))

meanCoeff <- rep(NA, times=length(colNumTest))

N <- rep(NA, times=length(colNumTest))

for (i in 1:length(colNumTest)){

test <- #abs(as.numeric(as.character(summaryData[,colNumTest[i]])) - as.numeric(as.character(summaryData[,colNumToCompare[i]])))

cor.test(as.numeric(as.character(summaryData[,colNumTest[i]])),

as.numeric(as.character( summaryData[,colNumToCompare[i]])), method="pearson")

barLower[i] <- test$conf.int[1]

barUpper[i] <- test$conf.int[2]

meanCoeff[i] <- test$estimate[1]

N[i] <- nrow(summaryData[!is.na(summaryData[,colNumTest[i]])&!is.na(summaryData[,colNumToCompare[i]]),])

}

plot(

x=0, y=0, xlab="", ylab="Coefficient of correlation",

xlim=c(0,length(meanCoeff)+1), ylim=c(0.6,1),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=length(meanCoeff), xintsmall=0.5, xintbig=1, ymin=0.6, ymax=1, yintsmall=0.025, yintbig=0.1, axisPlot=FALSE)

axis(side=2, at=seq(from=0.6, to=1, by=0.2), labels=seq(from=0.6, to=1, by=0.2), las=2, tcl=-0.25)

addLabel(x=0.05, y=0.075, label="A", radius=0.75, circle=TRUE, circle.bg="black", font.col="white")

#Comparison

whatCompared <- c(

rep("Brain", times=10),

"Hippocampus",

"Hippocampus",

"Hippocampus",

"Neocortex",

"Neocortex",

"Neocortex",

"Neocortex",

"Neocortex",

"Neocortex",

"Cerebellum",

"Cerebellum",

"Cerebellum",

"Striatum",

rep("Body", times=6),

"Frug.",

"Fol."

)

#Plot legend of what is compared in coloured rectangles

whereToPlot <- as.data.frame(table(whatCompared))

whereToPlot$loc <- whereToPlot$Freq/2

#Colour rectangle to indicate what is compared

refLoc=0

for (i in 1:length(whatCompared)){

rect(xleft=i-1,

xright=i,

ybottom=0.6-0.05\*0.4,#-0.05,

ytop=0.6,#0,

border=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])],

col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])],

xpd=TRUE

)

errorBars(location=i-0.5, meanPt=meanCoeff[i], refUnit=1, col="black", minValue=0, maxValue=1, upperBarValue=barUpper[i], lowerBarValue=barLower[i], horiz=FALSE, symmetrical=FALSE)

points(x=i-0.5, y=meanCoeff[i], pch=19, col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])],

xpd=TRUE)

text(x=i-0.5, y=barUpper[i]+0.015, labels=N[i], pch=19, col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])], cex=0.8,xpd=TRUE)

if(i==length(whatCompared)|whatCompared[i]!=whatCompared[i+1]){

refLoc=refLoc+whereToPlot$loc[whereToPlot[,1]==whatCompared[i]]

if(whereToPlot$loc[whereToPlot[,1]==whatCompared[i]]<=1){

segments(x0=refLoc, x1=refLoc, y0=0.6-0.025\*0.4,#-0.025,

y1=0.6-0.075\*0.4 + addToY[i]\*0.4,#-0.075,

col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])], xpd=TRUE)

text(x=refLoc, y=0.6-0.1\*0.4 + addToY[i]\*0.4,#-0.1,

labels=whereToPlot[whereToPlot[,1]==whatCompared[i],1], col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])], cex=cexText[i], xpd=TRUE)

}

else{

text(x=refLoc, y=0.6-0.025\*0.4,#-0.025,

labels=whereToPlot[whereToPlot[,1]==whatCompared[i],1], col="black", cex=cexText[i], xpd=TRUE)

}

refLoc=refLoc+whereToPlot$loc[whereToPlot[,1]==whatCompared[i]]#add second time for having complete rectangle

}

}

###------

### VARIABILITY

#cbind(colnames(summaryData[colNumTest]), colnames(summaryData[colNumToCompare]))

#Vectors to save results

barLower <- rep(NA, times=length(colNumTest))

barUpper <- rep(NA, times=length(colNumTest))

meanCoeff <- rep(NA, times=length(colNumTest))

N <- rep(NA, times=length(colNumTest))

for (i in 1:length(colNumTest)){

transitoryinit <- as.data.frame(cbind(as.numeric(as.character(summaryData[,colNumTest[i]])),as.numeric(as.character(summaryData[,colNumToCompare[i]]))))

transitoryinit <- transitoryinit[!is.na(transitoryinit[,1])&!is.na(transitoryinit[,2]),]

# transitoryinit <- transitoryinit/max(apply(transitoryinit, 2, max))

transitory <- apply(transitoryinit, 1, function(v){abs(v[1]\*\*2 - v[2]\*\*2)/2/v[1]/v[2]}) #note= (abs((v1-v2))/v1 + abs((v2-v1))/v2)/2

#When rate is 0 for both, gives NA, so to transform to 0

transitory[is.na(transitory)] <- 0

transitory[!is.finite(transitory)] <- apply(transitoryinit[!is.finite(transitory),], 1, function(x) abs(max(x)\*\*2 - 1\*\*2)/2/max(x)/1)

barLower[i] <- mean(transitory) - sd(transitory)/sqrt(length(transitory))#min(transitory)

barUpper[i] <- mean(transitory) + sd(transitory)/sqrt(length(transitory))#max(transitory)

meanCoeff[i] <- mean(transitory)

N[i] <- length(transitory)

}

ymax <- 1#round((barUpper)/10)\*10

plot(

x=0, y=0, xlab="", ylab="Variability",

xlim=c(0,length(meanCoeff)+1), ylim=c(0,ymax),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=length(meanCoeff), xintsmall=0.5, xintbig=1, ymin=0, ymax=ymax, yintsmall=0.05, yintbig=0.2, axisPlot=FALSE)

axis(side=2, at=seq(from=0, to=ymax, by=0.2), labels=seq(from=0, to=ymax, by=0.2), las=2, tcl=-0.25)

addLabel(x=0.05, y=0.075, label="B", radius=0.75, circle=TRUE, circle.bg="black", font.col="white")

#Comparison

whatCompared <- c(

rep("Brain", times=10),

"Hippocampus",

"Hippocampus",

"Hippocampus",

"Neocortex",

"Neocortex",

"Neocortex",

"Neocortex",

"Neocortex",

"Neocortex",

"Cerebellum",

"Cerebellum",

"Cerebellum",

"Striatum",

rep("Body", times=6),

"Frug.",

"Fol."

)

#Plot legend of what is compared in coloured rectangles

whereToPlot <- as.data.frame(table(whatCompared))

whereToPlot$loc <- whereToPlot$Freq/2

#Colour rectangle to indicate what is compared

refLoc=0

for (i in 1:length(whatCompared)){

rect(xleft=i-1,

xright=i,

ybottom=-0.05,

ytop=0,

border=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])],

col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])],

xpd=TRUE

)

errorBars(location=i-0.5, meanPt=meanCoeff[i], refUnit=1, col="black", minValue=0, maxValue=1, upperBarValue=barUpper[i], lowerBarValue=barLower[i], horiz=FALSE, symmetrical=FALSE)

points(x=i-0.5, y=meanCoeff[i], pch=19, col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])],

xpd=TRUE)

text(x=i-0.5, y=barUpper[i]+0.05, labels=N[i], pch=19, col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])], cex=0.8,xpd=TRUE)

if(i==length(whatCompared)|whatCompared[i]!=whatCompared[i+1]){

refLoc=refLoc+whereToPlot$loc[whereToPlot[,1]==whatCompared[i]]

if(whereToPlot$loc[whereToPlot[,1]==whatCompared[i]]<=1){

segments(x0=refLoc, x1=refLoc, y0=-0.025,

y1=-0.075 + addToY[i],

col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])], xpd=TRUE)

text(x=refLoc, y=-0.1 + addToY[i],

labels=whereToPlot[whereToPlot[,1]==whatCompared[i],1], col=colourWhatCompared[which(unique(whatCompared)==whatCompared[i])], cex=cexText[i], xpd=TRUE)

}

else{

text(x=refLoc, y=-0.025,

labels=whereToPlot[whereToPlot[,1]==whatCompared[i],1], col="black", cex=cexText[i], xpd=TRUE)

}

refLoc=refLoc+whereToPlot$loc[whereToPlot[,1]==whatCompared[i]]#add second time for having complete rectangle

}

}

##----

## Repeatability of measure

colRep <- list(c(4, 14, 17, 22, 23),

c(5, 15, 18),

c(6, 12, 16, 19),

c(7, 13, 20),

#c(8, 21),

c(24, 25, 26, 27)#,

#c(29, 33),

#c(30, 34)

)

ymax <- 1

plot(

x=0, y=0, xlab="", ylab="Repeatability",

xlim=c(0,length(colRep)+1), ylim=c(0,ymax),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

addGrid(xmin=0, xmax=length(colRep), xintsmall=0.5, xintbig=1, ymin=0, ymax=ymax, yintsmall=0.05, yintbig=0.2, axisPlot=FALSE)

axis(side=2, at=seq(from=0, to=ymax, by=0.2), labels=seq(from=0, to=ymax, by=0.2), las=2, tcl=-0.25)

addLabel(x=0.05\*2, y=0.075, label="C", radius=0.75\*length(colRep)/length(meanCoeff)\*2, circle=TRUE, circle.bg="black", font.col="white")

#Comparison

whatCompared2 <- c(

"Brain",

"Hippocampus",

"Neocortex",

"Cerebellum",

#"Striatum",

"Body"#,

#"Fol.",

#"Frug."

)

for(i in 1:length(colRep)){

dataRdc <- summaryData[, colRep[[i]]]

#Normalise by max for all species (i.e. by row)

maxVector <- apply(dataRdc, 1, function(x)max(x, na.rm=TRUE))

maxVector[!is.finite(maxVector)] <- NA

dataRdc <- apply(dataRdc, 2, function(x) x/maxVector)

dataRdc <- as.data.frame(dataRdc)

#Create common ID

dataRdc$id <- 1:nrow(dataRdc)

#Switch to 1 row=1value

dataRdc <- pivot\_longer(dataRdc, col=1:(ncol(dataRdc)-1), names\_to="Dataset", values\_to="Value")

#Remove NAs

dataRdc <- dataRdc[!is.na(dataRdc$Value),]

#Keep thos with multiple obs

whichKeep <- dataRdc %>% count(id)

whichKeep <- whichKeep$id[whichKeep$n > 2]

dataRdc <- dataRdc[dataRdc$id %in% whichKeep,]

dataRdc <- pivot\_wider(dataRdc, names\_from="Dataset", values\_from="Value")

withinVariance <- apply(dataRdc[,2:ncol(dataRdc)], 1, function(x) var(x, na.rm=TRUE))

betweenVariance <- var(apply(dataRdc[,2:ncol(dataRdc)], 1, function(x) mean(x, na.rm=TRUE)), na.rm=TRUE)

repeatability <- betweenVariance/(betweenVariance + withinVariance)

#Plot mean +/- SE

errorBars(location=i-0.5, meanPt=mean(repeatability),

refUnit=1, col="black", minValue=0, maxValue=1, upperBarValue=mean(repeatability)+sd(repeatability)/sqrt(length(repeatability)),

lowerBarValue=mean(repeatability)-sd(repeatability)/sqrt(length(repeatability)), horiz=FALSE, symmetrical=FALSE)

points(x=i-0.5, y=mean(repeatability), pch=19, col=colourWhatCompared[which(unique(whatCompared)==whatCompared2[i])],

xpd=TRUE)

text(x=i-0.5, y=mean(repeatability)+sd(repeatability)/sqrt(length(repeatability))+0.05, labels=length(repeatability), pch=19, col=colourWhatCompared[which(unique(whatCompared)==whatCompared2[i])], cex=0.8, xpd=TRUE)

# library(rptR)

# repeatabilityTest <- rpt(Value ~ (1 | id), grname = "id", data = dataRdc, datatype = "Proportion",

# nboot = 10, npermut = 0)

# Too few data to do that way

}

```

```{r figdata, fig.pos='H', include=TRUE, warning = FALSE, message = FALSE, fig.width=20, fig.height=25, fig.cap="Data availability | Black boxes indicate data availability while grey boxes indicate absence of data."}

#

# plot(

# x=0, y=0, xlab="", ylab="", cex.sub=1.6,

# xlim=c(-10,ncol(dataForSample)-1), ylim=c(0,nrow(dataForSample)),

# las=1, type="n", tcl=-0.25, frame.plot=FALSE,

# xaxt="n",xaxs="i",yaxs="i", yaxt="n")

#

# text(x=rep(-2, times=nrow(dataForSample)), y=1:nrow(dataForSample)-0.5, labels=dataForSample$SpeciesForPhylo, xpd=TRUE, cex=0.4)

# text(x=rep(-0, times=nrow(dataForSample)), y=1:nrow(dataForSample)-0.5, labels=dataForSample$Species, xpd=TRUE, cex=0.4)

# text(x=3:ncol(dataForSample)-1.5, y=rep(nrow(dataForSample)+3, times=length(3:ncol(dataForSample))), labels=colnames(dataForSample)[3:ncol(dataForSample)], xpd=TRUE, cex=0.4, srt=45)

#

# for(i in 1:nrow(dataForSample)){

# for(j in 3:ncol(dataForSample)){

# if(!is.na(dataForSample[i,j])&dataForSample[i,j]==1){

# rect(

# xleft=j-2,

# xright=j-1,

# ybottom=i-1,

# ytop=i,

# border="black",

# col="black"

# )

# } else{

# rect(

# xleft=j-2,

# xright=j-1,

# ybottom=i-1,

# ytop=i,

# border="lightgrey",

# col="lightgrey"

# )

# }

# }

# }

#

# addGrid(xmin=1, xmax=ncol(dataForSample), xintsmall=1, xintbig=1, ymin=0, ymax=nrow(dataForSample), yintsmall=1, yintbig=1, colsmall="white", colbig="white", axisPlot=FALSE)

#

#

#

# plot(

# x=0, y=0, xlab="", ylab="", cex.sub=1.6,

# xlim=c(-10,ncol(dataForSample)-1), ylim=c(0,nrow(dataForSample)),

# las=1, type="n", tcl=-0.25, frame.plot=FALSE,

# xaxt="n",xaxs="i",yaxs="i", yaxt="n")

#

# text(x=rep(-2, times=nrow(dataForSample)), y=1:nrow(dataForSample)-0.5, labels=dataForSample$SpeciesForPhylo, xpd=TRUE, cex=0.4)

# text(x=rep(-0, times=nrow(dataForSample)), y=1:nrow(dataForSample)-0.5, labels=dataForSample$Species, xpd=TRUE, cex=0.4)

# text(x=3:ncol(dataForSample)-1.5, y=rep(nrow(dataForSample)+3, times=length(3:ncol(dataForSample))), labels=colnames(dataForSample)[3:ncol(dataForSample)], xpd=TRUE, cex=0.4, srt=45)

#

# for(i in 1:nrow(dataForSample)){

# for(j in 3:ncol(dataForSample)){

# if(!is.na(dataForSample[i,j])&dataForSample[i,j]==1){

# rect(

# xleft=j-2,

# xright=j-1,

# ybottom=i-1,

# ytop=i,

# border="black",

# col="black"

# )

# } else{

# rect(

# xleft=j-2,

# xright=j-1,

# ybottom=i-1,

# ytop=i,

# border="lightgrey",

# col="lightgrey"

# )

# }

# }

# }

#

# addGrid(xmin=1, xmax=ncol(dataForSample), xintsmall=1, xintbig=1, ymin=0, ymax=nrow(dataForSample), yintsmall=1, yintbig=1, colsmall="white", colbig="white", axisPlot=FALSE)

dataForSample <- dataForSample[dataForSample$SpeciesForPhylo != "Homo\_sapiens",]

dataForSample <- dataForSample[order(dataForSample$SpeciesForPhylo),]

nbPlot=4

layout(mat=t(c(1:nbPlot)), widths=rep(5, times=nbPlot), heights=c(5\*nbPlot))

par(mar=c(0, 0, 3, 0.5), mgp=c(2, 0.5, 0), xpd=TRUE)

for(p in 1:nbPlot){

dataForSample\_rdcplot <- dataForSample[(1+(p-1)\*nrow(dataForSample)/nbPlot):(1+(p)\*nrow(dataForSample)/nbPlot),]

plot(

x=0, y=0, xlab="", ylab="", cex.sub=1.6,

xlim=c(-10,ncol(dataForSample\_rdcplot)-1), ylim=c(0,nrow(dataForSample\_rdcplot)+3),

las=1, type="n", tcl=-0.25, frame.plot=FALSE,

xaxt="n",xaxs="i",yaxs="i", yaxt="n")

text(x=rep(-5, times=nrow(dataForSample\_rdcplot)), y=1:nrow(dataForSample\_rdcplot)-0.5, labels=dataForSample\_rdcplot$SpeciesForPhylo, xpd=TRUE, cex=1.1)

text(x=rep(-0.5, times=nrow(dataForSample\_rdcplot)), y=1:nrow(dataForSample\_rdcplot)-0.5, labels=dataForSample\_rdcplot$Species, xpd=TRUE, cex=1.1)

text(x=3:ncol(dataForSample\_rdcplot)-1.5, y=rep(nrow(dataForSample\_rdcplot)+2, times=length(3:ncol(dataForSample\_rdcplot))), labels=colnames(dataForSample\_rdcplot)[3:ncol(dataForSample\_rdcplot)], xpd=TRUE, cex=1.1, srt=45)

for(i in 1:nrow(dataForSample\_rdcplot)){

for(j in 3:ncol(dataForSample\_rdcplot)){

if(!is.na(dataForSample\_rdcplot[i,j])&dataForSample\_rdcplot[i,j]==1){

rect(

xleft=j-2,

xright=j-1,

ybottom=i-1,

ytop=i,

border="black",

col="black"

)

} else{

rect(

xleft=j-2,

xright=j-1,

ybottom=i-1,

ytop=i,

border="lightgrey",

col="lightgrey"

)

}

}

}

addGrid(xmin=1, xmax=ncol(dataForSample\_rdcplot), xintsmall=1, xintbig=1, ymin=0, ymax=nrow(dataForSample\_rdcplot), yintsmall=1, yintbig=1, colsmall="white", colbig="white", axisPlot=FALSE)

#if(p==2){cat('\r\n\r\n')} #Allows to consider plot as a new figure at each step of the loop

}

```

\clearpage

### Primate diversification rate over time

```{r figdiversificationTime, fig.pos='H', include=TRUE, warning = FALSE, message = FALSE, fig.width=4.5, fig.height=4.5, fig.cap="\\footnotesize Diversification rate over time in the Primate taxon | The average diversification rate estimated based on an assumed sampling fraction of primate species ranging from 60 to 90% (at a step of 10%) is depicted by the plain line. The grey background depicts the standard deviation. The two breakpoints, depicted by the plain dots and the vertical dotted bars, were thus estimated based on a three-linear regression segmentation using the \*strucchange\* package [@strucchange1; @strucchange2; @strucchange3; see the vignette package for statistical details]. The three fitted regressions are displayed by the dashed lines. The choice of two breakpoints was priorily assessed by choosing the number of breakpoints minimizing the the Bayesian Information Criterion. The identified breakpoints coin with identified sharp decrease in extinction rate [@arbour2017major; @springer2012macroevolutionary] due to the emergence of more favourable environmental conditions stemming from a progressive warming after harsh temperature cooling that started earlier in the Oligocene until reaching a mid-Miocene Climatic Optimum [@fleagle2006biogeography]."}

##Plot diversification in function of time

xmin=floor(min(aggregatedSpeciationTime.mean[,1]/10))\*10

xmax=ceiling(max(aggregatedSpeciationTime.mean[,1]/10))\*10

ymin=floor(min(aggregatedSpeciationTime.mean[,2]\*10))/10

ymax=ceiling(max(aggregatedSpeciationTime.mean[,2]\*10))/10

plot(0, 0, xlab="Time before present (Myr)", ylab="Diversification rate",

xlim=c(xmin, xmax), ylim=c(ymin, ymax),

las=1, type="n", tcl=-0.25, bty="n",

xaxt="n",xaxs="i",yaxs="i", yaxt="n",

xpd=TRUE)

#Add grid

addGrid(

xmin=xmin, xmax=xmax, xintsmall=(xmax-xmin)/20, xintbig=(xmax-xmin)/5,

ymin=ymin, ymax=ymax, yintsmall=(ymax-ymin)/20, yintbig=(ymax-ymin)/5,

axisPlot=TRUE, round=TRUE, digit=c(2,2))

axis(side=1, at=round(seq(from=xmin, to=xmax, by=(xmax-xmin)/5), digit=1), labels=round(seq(from=xmin, to=xmax, by=(xmax-xmin)/5), digit=1), las=1, tcl=-0.25)

#Add background se

polygon(

x=c(aggregatedSpeciationTime.mean[,1], rev(aggregatedSpeciationTime.mean[,1])),

y=c(aggregatedSpeciationTime.mean[,2]-aggregatedSpeciationTime.sd[,2], rev(aggregatedSpeciationTime.mean[,2]+aggregatedSpeciationTime.sd[,2])),

col=grey(level=0.5, alpha=0.15),

border=NA

)

#Add mean

lines(aggregatedSpeciationTime.mean[,1], aggregatedSpeciationTime.mean[,2])

library(strucchange)

## confidence interval

colnames(aggregatedSpeciationTime.mean) <- c("Time", "Diversification")

yFirst <- aggregatedSpeciationTime.mean$Diversification[which((abs(aggregatedSpeciationTime.mean$Time+dateFirstRupt[2]))==min(abs(aggregatedSpeciationTime.mean$Time+dateFirstRupt[2])))]

ySecond <- aggregatedSpeciationTime.mean$Diversification[which((abs(aggregatedSpeciationTime.mean$Time+dateSecondRupt[2]))==min(abs(aggregatedSpeciationTime.mean$Time+dateSecondRupt[2])))]

# Rupture points and (CI too reduced to be plotted)

points(c(-dateFirstRupt[2], -dateSecondRupt[2]), c(yFirst, ySecond), pch=19)

# errorBars(location=c(yFirst),

# meanPt=c(-dateFirstRupt[2]),

# barValue=c(0,0), refUnit=1,

# minValue=-80, maxValue=80,

# upperBarValue=c(-dateFirstRupt[1]), lowerBarValue=c(-dateFirstRupt[3]),

# col="black", lty=1,

# horiz=TRUE, symmetrical=FALSE)

#Vertical bars

segments(

x0=c(-dateFirstRupt[2], -dateSecondRupt[2]),

x1=c(-dateFirstRupt[2], -dateSecondRupt[2]),

y0=c(0,0),

y1=c(yFirst, ySecond),

lty=3

)

mtext(side=1, at=c(-dateFirstRupt[2], -dateSecondRupt[2]), line=0, text=c(-dateFirstRupt[2], -dateSecondRupt[2]), cex=0.8)

#three fitted regressions

fm1 <- lm(Diversification ~ breakfactor(bp.resp, breaks = 2)\*Time, data=aggregatedSpeciationTime.mean)

reg1 <- summary(fm1)$coefficients[1,1] + aggregatedSpeciationTime.mean$Time\*summary(fm1)$coefficients[4,1]

reg2 <- summary(fm1)$coefficients[1,1] + summary(fm1)$coefficients[2,1] + aggregatedSpeciationTime.mean$Time\*(summary(fm1)$coefficients[4,1] + summary(fm1)$coefficients[5,1])

reg3 <- summary(fm1)$coefficients[1,1] + summary(fm1)$coefficients[3,1] + aggregatedSpeciationTime.mean$Time\*(summary(fm1)$coefficients[4,1] + summary(fm1)$coefficients[6,1])

lines(aggregatedSpeciationTime.mean$Time, reg1, lty = 2)

lines(aggregatedSpeciationTime.mean$Time[aggregatedSpeciationTime.mean$Time > -40], reg2[aggregatedSpeciationTime.mean$Time > -40], lty = 2)#truncaturate for readability

lines(aggregatedSpeciationTime.mean$Time[aggregatedSpeciationTime.mean$Time > -16], reg3[aggregatedSpeciationTime.mean$Time > -16], lty = 2)#truncaturate for readability

```

## Phylogenetic regressions: results, stability and assumption

### Model results

<!-- We present below the visual fit of phylogenetic regressions. -->

(a) Phylogenetic regressions: selection gradient

\clearpage

```{=latex}

\begin{figure}

\centering\includegraphics[width=0.65\linewidth]{C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Plots/selectionGradientPGLS.pdf}

\caption{\scriptsize{Phylogenetic regressions of relative brain size in function of sympatry intensity indices | Left graphics depict the effect of the number of sympatric species on the brain size, when the effect of the percentage of the distribution range overlapped by sympatric species is averaged, while the right graphics do the opposite. Raw data are depicted with points, while the segments that link them correspond to the projected phylogenetic tree. The model fit is shown with the plain black line and the associated 95\% confidence interval is depicted by the transparent gray background.}}\label{fig:figRegressionGradient}

\end{figure}

```

\clearpage

(b) Phylogenetic regressions: diversification and brain size

```{=latex}

\begin{figure}

\centering\includegraphics[width=0.7\linewidth]{C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Plots/diversificationPGLS.pdf}

\caption{\scriptsize{Phylogenetic regressions of the diversification rate in function of the size of the different brain areas | Raw data are depicted with points, while the segments that link them correspond to the projected phylogenetic tree. The model fit is shown with the plain black line and the associated 95\% highest density posterior is depicted by the transparent gray background.}}\label{fig:figRegressionDiversification}

\end{figure}

```

\clearpage

(b) Phylogenetic regressions: diversification and sympatry

```{=latex}

\begin{figure}

\centering\includegraphics[width=0.65\linewidth]{C:/Users/robira/Documents/PhD/Meta\_analysis/Meta\_analysis\_cognition\_primates/Plots/diversificationAndSympatryPGLS.pdf}

\caption{\scriptsize{Phylogenetic regressions of diversification rate in function of sympatry intensity indices | Left graphics depict the effect of the number of sympatric species on the brain size, when the effect of the percentage of the distribution range overlapped by sympatric species is averaged, while the right graphics do the opposite. Raw data are depicted with points, while the segments that link them correspond to the projected phylogenetic tree. The model fit is shown with the plain black line and the associated 95\% confidence interval is depicted by the transparent gray background.}}\label{fig:figRegressionDiversificationSympatry}

\end{figure}

```

\clearpage

(d) Forest plot of estimates

```{r forestPlot, fig.pos='H', include=TRUE, warning = FALSE, message = FALSE, fig.width=6.25, fig.height=6.25, fig.cap="Forest plot of the phylogenetic regressions | CI: Confindence Interval, HDP: Highest Posterior Density (when brain size is the predictor, they are barely visible because reduced). Plain dots depict negative effects, open dots depict positive effects. "}

source("T:/Saved\_PhD/Empirical\_analysis/Scripts&Functions/Functions/toolbox.R")

par(mar=c(8, 8, 0.5, 2), mgp=c(2.5, 0.5, 0), xpd=TRUE)

emptyPlot(xlim=c(-3,3), ylim=c(-3,3), asp=1)

#Grid

addGrid(

cexAxisX=1.15, cexAxisY=1.15,

xmin=-3, xmax=3, xintsmall=0.1, xintbig=0.5,

ymin=-3, ymax=3, yintsmall=0.1, yintbig=0.5,

axisPlot=FALSE, round=TRUE, digit=c(2,2), contour=TRUE)

#0 line

segments(x0=0, x1=0, y0=-3, y1=3, lty=2)

#Xaxis

axis(side=1, at=seq(-3, 3, 0.5), pos=-3, labels=seq(-3, 3, 0.5), las=1, tcl=-0.25)

mtext(side=1, at=0, line=1, text="Model estimate (point) and 95% CI/HDP (segment)")

#yAxis

axis(side=2, at=c(2, 0, -2), pos=-3, labels=c("Overlap (%)", "N sympatric\nspecies", "Relative\nbrain size"), las=1, tcl=-0.25)

mtext(side=2, at=0, line=6, text="PARAMETER", font=2)

#Add the different values

#Analysis of brain size and sympatry

traitName\_rdc=c("EQ (log)", #"Brain (/bodymass, log)",

"Striatum (/bodymass, log)",

"MOB (/bodymass, log)",

"Hippocampus (/bodymass, log)",

"Neocortex (/bodymass, log)",

"Cerebellum (/bodymass, log)",

"DiversificationAndSympatry"

)

traitName\_legend <- c("EQ", "Striatum", "MOB", "Hippocampus", "Neocortex", "Cerebellum", "Diversification")

library(RColorBrewer)

colourModelsReg <- c("darkgrey", brewer.pal(n = 5, name = "Set1")[1:5], "black")

for(a in 1:length(traitName\_rdc)){

if(a!=length(traitName\_rdc)){

model <- get(paste("modelBrain", traitName\_rdc[a], sep="\_"))

}else{

model <- get(paste("modelBrain", traitName\_rdc[a], sep=""))

}

#Overlap

errorBars(location=2.5-(a-1)\*1/length(traitName\_rdc),

meanPt=cbind(model$bootmean, t(model$bootconfint95))[2,1],

barValue=1,

refUnit=0.75,

minValue=-3,

maxValue=3,

upperBarValue=cbind(model$bootmean, t(model$bootconfint95))[2,2],

lowerBarValue=cbind(model$bootmean, t(model$bootconfint95))[2,3],

col=colourModelsReg[a],

lty=1,

horiz=TRUE,

symmetrical=FALSE)

points(x=cbind(model$bootmean, t(model$bootconfint95))[2,1], y=2.5-(a-1)\*1/length(traitName\_rdc), pch=ifelse(cbind(model$bootmean, t(model$bootconfint95))[2,1]>0,21,19), bg="white", col=colourModelsReg[a])

if(a!=length(traitName\_rdc)){

text(x=cbind(model$bootmean, t(model$bootconfint95))[2,3], y=2.5-(a-1)\*1/length(traitName\_rdc),

pos=4, col=colourModelsReg[a],

labels=pvalueToText(as.numcharac(results.df\_gradient[grep(traitName\_legend[a], results.df\_gradient[,1]) + 2,7])), cex=0.75)

}else{

text(x=cbind(model$bootmean, t(model$bootconfint95))[2,3], y=2.5-(a-1)\*1/length(traitName\_rdc),

pos=4, col=colourModelsReg[a],

labels=pvalueToText(as.numcharac(results.df\_diversificationAndSympatry[3,7])), cex=0.75)

}

#N species

errorBars(location=0.5-(a-1)\*1/length(traitName\_rdc),

meanPt=cbind(model$bootmean, t(model$bootconfint95))[3,1],

barValue=1,

refUnit=0.75,

minValue=-3,

maxValue=3,

upperBarValue=cbind(model$bootmean, t(model$bootconfint95))[3,2],

lowerBarValue=cbind(model$bootmean, t(model$bootconfint95))[3,3],

col=colourModelsReg[a],

lty=1,

horiz=TRUE,

symmetrical=FALSE)

points(x=cbind(model$bootmean, t(model$bootconfint95))[3,1], y=0.5-(a-1)\*1/length(traitName\_rdc), pch=ifelse(cbind(model$bootmean, t(model$bootconfint95))[3,1]>0,21,19), bg="white", col=colourModelsReg[a])

if(a!=length(traitName\_rdc)){

text(x=cbind(model$bootmean, t(model$bootconfint95))[3,3], y=0.5-(a-1)\*1/length(traitName\_rdc),

pos=4, col=colourModelsReg[a],

labels=pvalueToText(as.numcharac(results.df\_gradient[grep(traitName\_legend[a], results.df\_gradient[,1]) + 3,7])), cex=0.75)

}else{

text(x=cbind(model$bootmean, t(model$bootconfint95))[3,3], y=0.5-(a-1)\*1/length(traitName\_rdc),

pos=4, col=colourModelsReg[a],

labels=pvalueToText(as.numcharac(results.df\_diversificationAndSympatry[4,7])), cex=0.75)

}

#Analysis of diversification and brain size

if(a!=length(traitName\_rdc)){

model <- get(paste("modelBrainDiversification", traitName[a], sep="\_"))

errorBars(location=-1.5-(a-1)\*1/(length(traitName\_rdc)-1),

meanPt=as.data.frame(summary(model)$solutions[,1:3])[2,1],

barValue=1,

refUnit=0.75,

minValue=-3,

maxValue=3,

upperBarValue=as.data.frame(summary(model)$solutions[,1:3])[2,2],

lowerBarValue=as.data.frame(summary(model)$solutions[,1:3])[2,3],

col=colourModelsReg[a],

lty=1,

horiz=TRUE,

symmetrical=FALSE)

points(x=as.data.frame(summary(model)$solutions[,1:3])[2,1], y=-1.5-(a-1)\*1/(length(traitName\_rdc)-1), pch=ifelse(as.data.frame(summary(model)$solutions[,1:3])[2,1]>0,21,19), bg="white", col=colourModelsReg[a])

text(x=as.data.frame(summary(model)$solutions[,1:3])[2,3], y=-1.5-(a-1)\*1/(length(traitName\_rdc)-1),

pos=4, col=colourModelsReg[a],

labels=pvalueToText(as.numcharac(results.df\_diversification[grep(traitName\_legend[a], results.df\_diversification[,1]) + 2,5])), cex=0.75)

}

}

legend(x=3, y=0.5,

pch=c(19,21),

bg=c("white", "white"),

col=c("black", "black"),

legend=c("> 0", "< 0"),

#title="Output:",

bty="n",

xjust=1,

xpd=TRUE)

legend(x=-2.5, y=-4,

pch=rep(19, times=length(traitName\_rdc)),

col=c(colourModelsReg),

legend=traitName\_legend,

#title="Output:",

ncol=3,

bty="n",

xpd=TRUE)

```

\clearpage

### Model stability

We present below statistical indicators related to changes in estimates when re-fitting the model considering sub-samples (i.e. DfBetas and Cook's distance), as well as when accounting for data variability (i.e. re-sampling among possible values given all datasets) or when using different parameterisation (i.e. "sampling fraction" of known species for diversification analysis).

```{r correctSensitivityTable}

#Include true estimator in the min-max range

#1)Change min-max interval

summarySensitivityGradient[,1:3+2] <- t(apply(summarySensitivityGradient[,1:3+2], 1, function(x){

x <- as.numeric(x)

if(x[1] > x[2]){

x[1] <- x[2]

} else if (x[2] > x[3]){

x[3] <- x[2]

}else{}

return(x)

}

)

)

summarySensitivityGradient[,4:6+2] <- t(apply(summarySensitivityGradient[,4:6+2], 1, function(x){

x <- as.numeric(x)

if(x[1] > x[2]){

x[1] <- x[2]

} else if (x[2] > x[3]){

x[3] <- x[2]

}else{}

return(x)

}

)

)

#2) Re-round intelligent with scientific writing

summarySensitivityGradient[, 1:6+2] <- t(apply(summarySensitivityGradient[, 1:6+2], 1, function(x)

roundIntelligent(as.numcharac(x))))

#1)Change min-max interval

summarySensitivityDiversification[,1:3+2] <- t(apply(summarySensitivityDiversification[,1:3+2], 1, function(x){

x <- as.numeric(x)

if(x[1] > x[2]){

x[1] <- x[2]

} else if (x[2] > x[3]){

x[3] <- x[2]

}else{}

return(x)

}

)

)

summarySensitivityDiversification[,4:6+2] <- t(apply(summarySensitivityDiversification[,4:6+2], 1, function(x){

x <- as.numeric(x)

if(x[1] > x[2]){

x[1] <- x[2]

} else if (x[2] > x[3]){

x[3] <- x[2]

}else{}

return(x)

}

)

)

summarySensitivityDiversification[,7:9+2] <- t(apply(summarySensitivityDiversification[,7:9+2], 1, function(x){

x <- as.numeric(x)

if(x[1] > x[2]){

x[1] <- x[2]

} else if (x[2] > x[3]){

x[3] <- x[2]

}else{}

return(x)

}

)

)

#2) Re-round intelligent with scientific writing

summarySensitivityDiversification[,-c(1,2)] <- t(apply(summarySensitivityDiversification[,-c(1,2)], 1, function(x) roundIntelligent(as.numcharac(x))))

```

\hfill

(a) Phylogenetic regressions: selection gradient

```{r tabledfsensitivity, include=TRUE}

summarySensitivityGradient <- summarySensitivityGradient[-(1:4),]

rownames(summarySensitivityGradient) <- NULL

knitr::kable(summarySensitivityGradient, escape=TRUE, booktabs = TRUE, #Remove brain 1:3

caption = "Sensitivity analysis of phylogenetic regressions to assess the relationship between speies sympatry and relative brain sizes | Depicted is the minimum and maximum of estimates when one observation was removed at a time (DfBetas) or when varying the used phylogenetic tree and the data sampling (Phylogeny/Data).") %>%

kableExtra::kable\_styling(latex\_options = "striped") %>%

kableExtra::kable\_styling(latex\_options="scale\_down") %>%

kableExtra::kable\_styling(latex\_options = "HOLD\_position") %>%

kableExtra::add\_header\_above(c("Regression" = 2, "DfBetas" = 3, "Phylogeny/Data" = 3))

```

\newpage

(b) Phylogenetic regressions: diversification and brain size

```{r tabledfsensitivity2, include=TRUE}

summarySensitivityDiversification <- summarySensitivityDiversification[-(1:3),]

rownames(summarySensitivityDiversification) <- NULL

knitr::kable(summarySensitivityDiversification[,-c(3,4,5)], escape=TRUE, booktabs = TRUE, #Remove brain 1:3

caption = "Sensitivity analysis of phylogenetic regressions to detect the assess the relationship between species diversifition and relative brain sizes| Depicted is the minimum and maximum of estimates when varying the used phylogenetic tree and the data sampling (Phylogeny/Data), or when the sampling fraction varied (Sampling fraction).") %>%

kableExtra::kable\_styling(latex\_options = "striped") %>%

kableExtra::kable\_styling(latex\_options="scale\_down") %>%

kableExtra::kable\_styling(latex\_options = "HOLD\_position") %>%

kableExtra::add\_header\_above(c("Regression" = 2, "Phylogeny/Data" = 3, "Sampling fraction" = 3))

```

\newpage

(b) Phylogenetic regressions: diversification and sympatry

```{r tabledfsensitivity3, include=TRUE}

knitr::kable(summarySensitivitydiversificationAndSympatry, escape=TRUE, booktabs = TRUE,

caption = "Sensitivity analysis of phylogenetic regressions to assess the selection diversificationAndSympatry direction | Depicted is the minimum and maximum of estimates when one observation was removed at a time (DfBetas) or when varying the used phylogenetic tree and the data sampling (Phylogeny/Data)") %>%

kableExtra::kable\_styling(latex\_options = "striped") %>%

kableExtra::kable\_styling(latex\_options="scale\_down") %>%

kableExtra::add\_header\_above(c("Model:" = 2, "DfBetas" = 3, "Phylogeny/Data" = 3, "Sampling fraction"))

```

\newpage

### Model assumptions

We present below the visual assessment of linear modelling assumptions (histogram of residuals, Q-Q plot, and scatterplot of fitted values vs residuals).

(a) Phylogenetic regressions: selection gradient

```{r modelAssumption, include=TRUE, warning = FALSE, message = FALSE, fig.width=7, fig.height=4.5, fig.cap="Model assumption check 'Brain size and sympatry' | Depicted are the histogram of residuals, the Q-Q plot, and the scatter plot of the fitted values \*vs\* the residuals."}

for(a in 1:length(traitName)){

if(a!=2){#Remove brain

model <- get(paste("modelBrain", traitName[a], sep="\_"))

diagnostics.plot(model)

text(x=0.85, y=0.15, paste("Model:\n",traitName[a], sep=""), xpd=TRUE)

if(a/2==floor(a/2)){

cat('\n') #Break page for new figure

}

}

}

```

\newpage

(b) Phylogenetic regressions: diversification and brain size

```{r modelAssumption2a, include=TRUE, warning = FALSE, message = FALSE, fig.width=7, fig.height=4.5, fig.cap="Model assumption check 'Diversity and brain size' | Trace and density of posteriors"}

for(a in 1:length(traitName)){

if(a!=2){#Remove brain

model <- get(paste("modelBrainDiversification", traitName[a], sep="\_"))

plot(model1$Sol)

mtext(paste("Fixed effects: ", traitName[a], sep=""), side = 3, line = -1, outer = TRUE, xpd=TRUE)

#cat('\n') #Break page for new figure

plot(model1$VCV)

mtext(paste("Random/residuals: ", traitName[a], sep=""), side = 3, line = -1, outer = TRUE, xpd=TRUE)

#text(x=max(fitted(model)) + abs(min(fitted(model))), y=mean(residuals(model)), paste("Model:\n",traitName[a], sep=""), xpd=TRUE)

if(a/2==floor(a/2)){

cat('\n') #Break page for new figure

}

}

}

```

```{r modelAssumption2b, include=TRUE, warning = FALSE, message = FALSE, fig.width=7, fig.height=10, fig.cap="Model assumption check 'Diversity and brain size' | Q-Q plot of the posterior distribution and the expected Gaussian distribution"}

layout(mat=matrix(1:(2\*ceiling(length(traitName)/2)), ncol=2), widths=c(5,5), heights=rep(5, times=2\*ceiling(length(traitName)/2)))

par(mar=c(3, 3, 3, 1), mgp=c(2, 0.5, 0), xpd=TRUE)

for(a in 1:length(traitName)){

if(a!=2){#Remove brain

model <- get(paste("modelBrainDiversification", traitName[a], sep="\_"))

posterior <- as.data.frame(model$Sol)

qqnorm(posterior$Trait, main=paste("Normal Q-Q Plot of the posterior distribution;\nModel:", traitName[a], sep=" "))

qqline(posterior$Trait)

#text(x=max(fitted(model)) + abs(min(fitted(model))), y=mean(residuals(model)), paste("Model:\n",traitName[a], sep=""), xpd=TRUE)

#cat('\n') #Break page for new figure

}

}

```

\newpage

(b) Phylogenetic regressions: diversification and sympatry

```{r modelAssumption3, include=TRUE, warning = FALSE, message = FALSE, fig.width=7, fig.height=4.5, fig.cap="Model assumption check 'Diversity and sympatry'| Depicted are the histogram of residuals, the Q-Q plot, and the scatter plot of the fitted values \*vs\* the residuals."}

model <- get(paste("modelBrainDiversificationAndSympatry", sep="\_"))

diagnostics.plot(model)

```

<!-- TC:endignore -->