

# Monitor Lizard Diversification with DAISIE

*Ian G. Brennan & Carlos Pavon Vazquez*

*16 June, 2020*

## Contents

<b>Read Me</b>	<b>1</b>
<b>Data Visualization</b>	<b>2</b>
<b>Data Preparation</b>	<b>8</b>
Extracting branching times . . . . .	8
Making a DAISIE datatable . . . . .	8
Making a DAISIE datalist . . . . .	9
<b>Fitting DAISIE models</b>	<b>10</b>
Equal Rates model with Diversity Dependence . . . . .	10
Equal Rates model without Diversity Dependence . . . . .	10
Two Rate Speciation model with Diversity Dependence . . . . .	10
Two Rate Speciation and Extinction model with Diversity Dependence . . . . .	11
Comparing the fit of DAISIE Models . . . . .	11
<b>Fitting DAISIE Models in Parallel</b>	<b>12</b>
Equal Rates model with Diversity Dependence . . . . .	12
Two Speciation Rate model with Diversity Dependence . . . . .	12
Comparing the fit of DAISIE Models run in parallel . . . . .	13
<b>Diversification Models with ClaDS</b>	<b>14</b>
ClaDS0—Pure Birth Model . . . . .	14
ClaDS2—Constant Turnover Model . . . . .	14

## Read Me

This is a living markdown document that will hopefully walk us through the data and code necessary to repeat the analyses of monitor lizard diversification using DAISIE. All the files and code are available at the GitHub Repository [MonitorDiversification](#) (this doesn't exist yet, just on my computer).

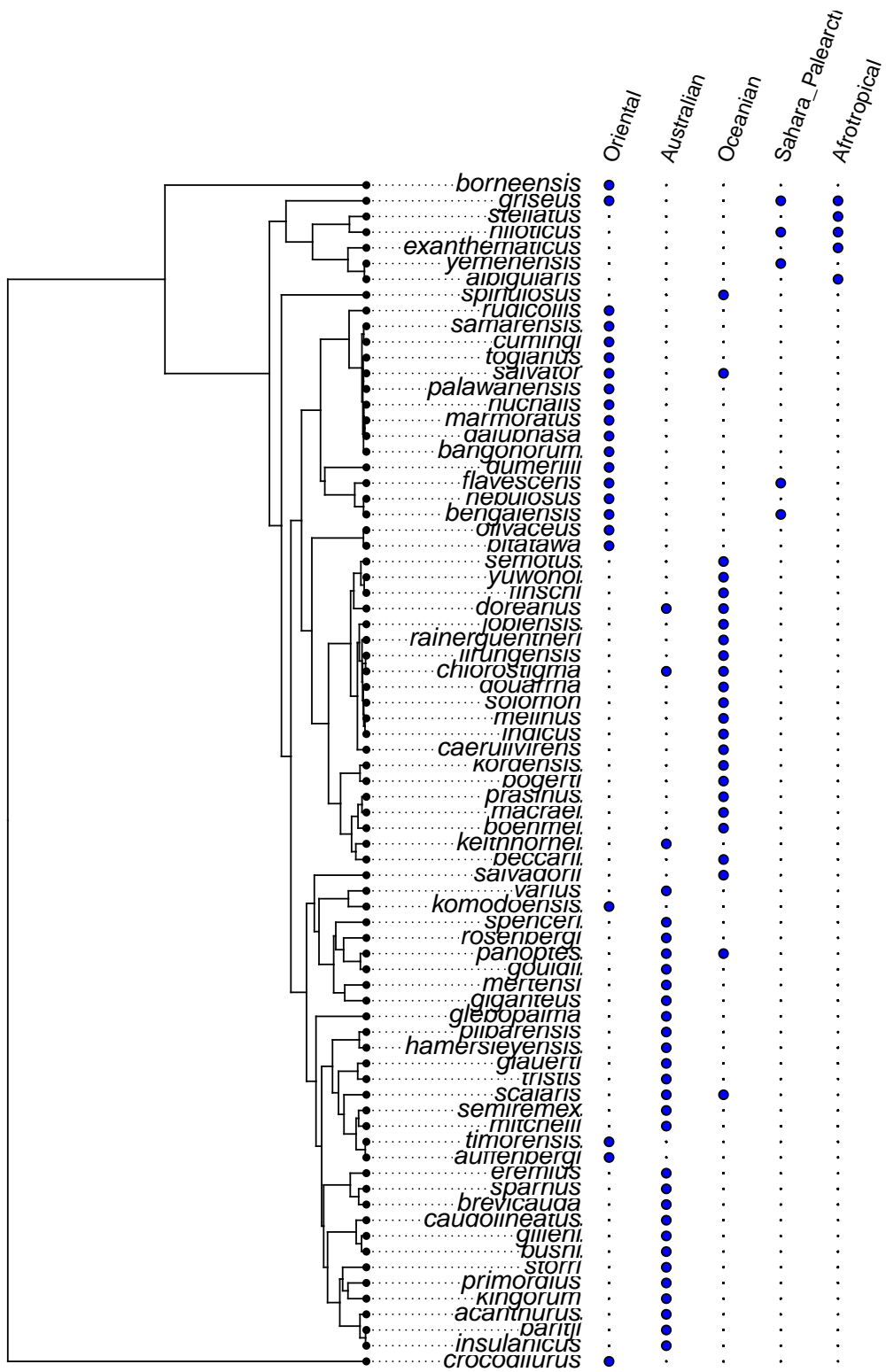
Start off by loading a few packages that we'll need along the way.

```
library(dplyr)
library(treeplyr)
library(RCurl)
library(phytools)
library(RColorBrewer)
library(ggplot2)
library(DAISIE)
# remember 'plyr' and 'dplyr' conflict, so don't load 'plyr'
```

## Data Visualization

Start by reading in the tree and visualize the distribution of taxa

```
vtree <- read.tree("UsedTree.tre");
regions <- read.delim("geo_file_Regions.txt", sep="\t", row.names = 1)
dotTree(vtree, regions, labels=T,length=8)
```



.....●●●●●

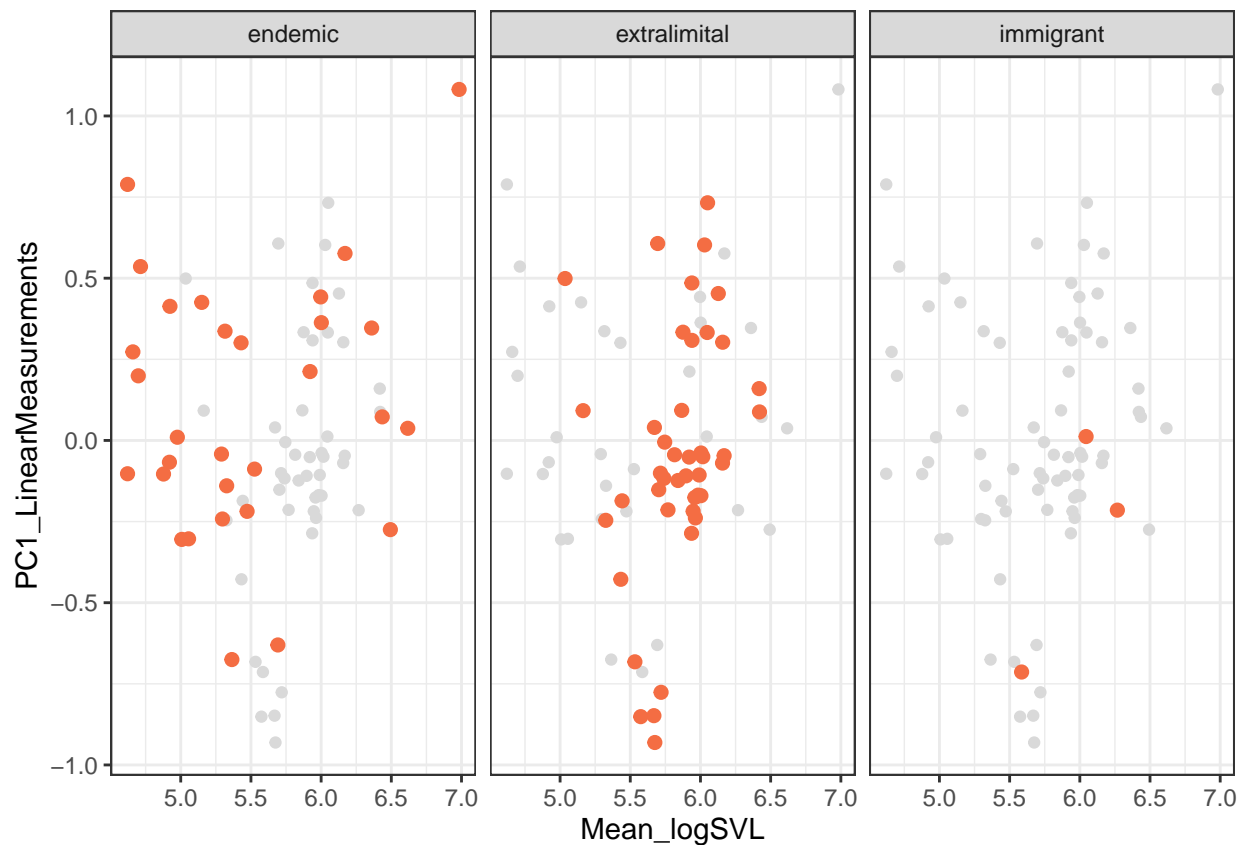
0 1

Read in the morphological data we have and visualize data.

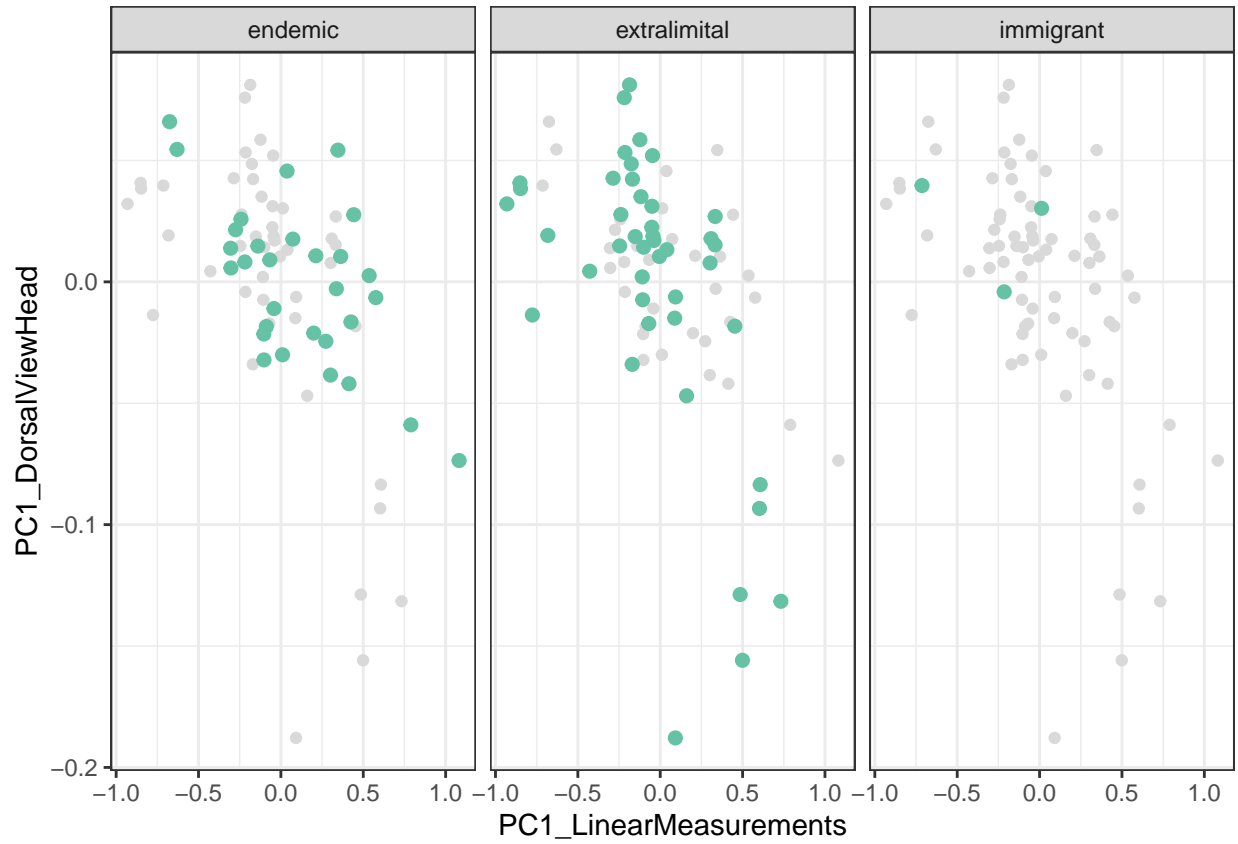
I've made an additional data column to look at the Australian monitor data. I've categorized them as "endemic" to Australia (most of the *Odatia/Varanus* subgenera), "immigrant" (*dore-anus/chlorostigma/keithornei*), or "extralimital" which is just all things existing elsewhere exclusively.

```
morph.data <- read.csv("MorphForIan.csv", header=T)
```

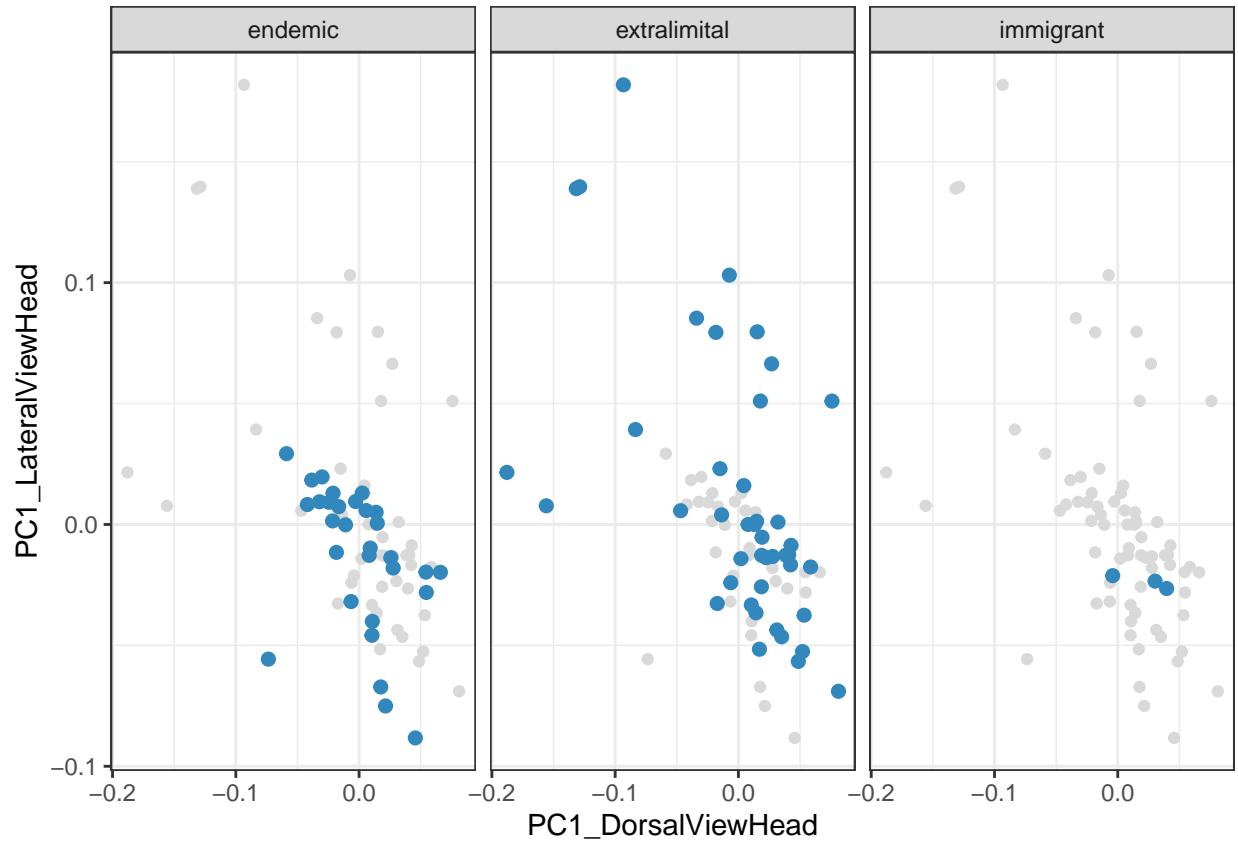
```
ggplot(morph.data, aes(Mean_logSVL, PC1_LinearMeasurements)) +  
  geom_point(data = transform(morph.data[c("Mean_logSVL",  
                                           "PC1_LinearMeasurements")]),  
            color = "grey85") +  
  geom_point(color = "#F46D43", size = 2) +  
  #stat_ellipse() +  
  theme_bw() +  
  facet_wrap(vars(Oz_Group))
```



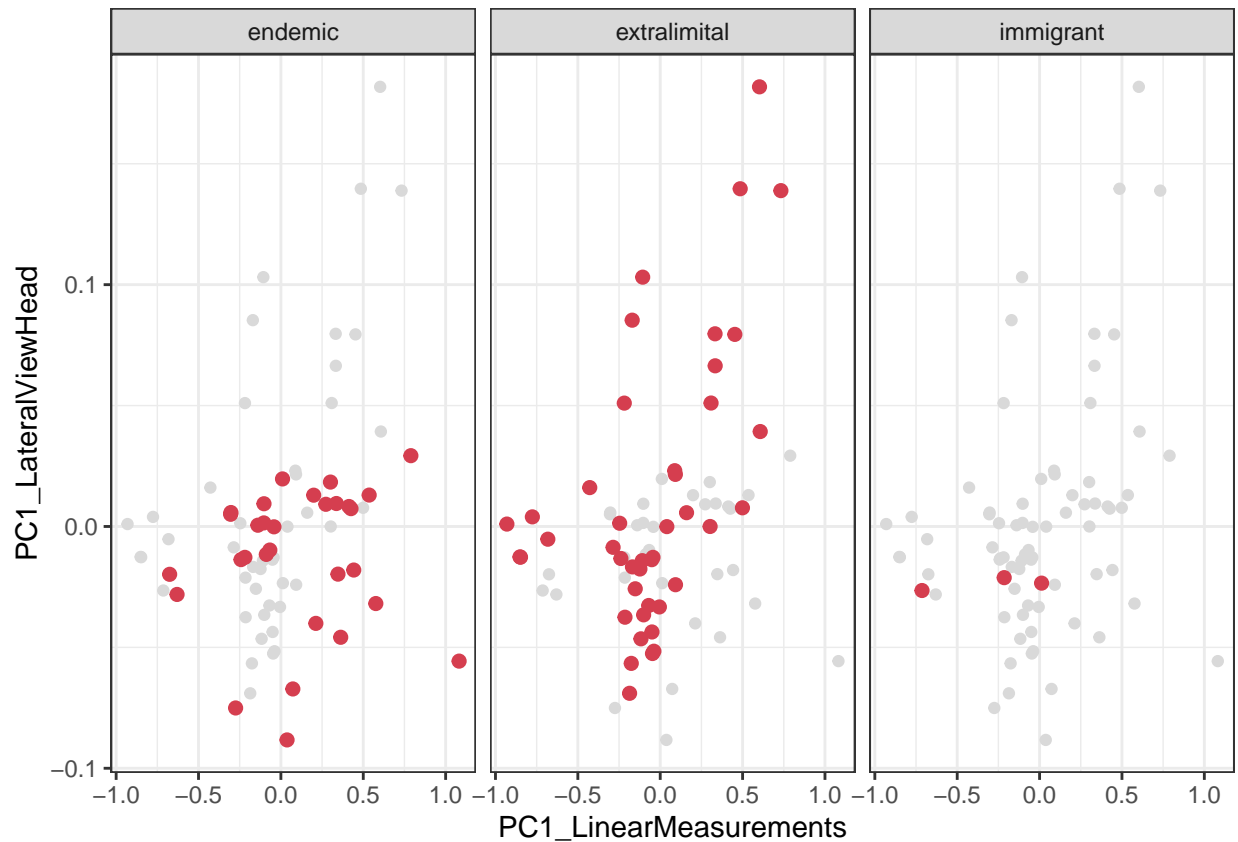
```
ggplot(morph.data, aes(PC1_LinearMeasurements, PC1_DorsalViewHead)) +
  geom_point(data = transform(morph.data[c("PC1_LinearMeasurements",
                                           "PC1_DorsalViewHead")]),
            color = "grey85") +
  geom_point(color = "#66C2A5", size = 2) +
  #stat_ellipse() +
  theme_bw() +
  facet_wrap(vars(Oz_Group))
```



```
ggplot(morph.data, aes(PC1_DorsalViewHead, PC1_LateralViewHead)) +
  geom_point(data = transform(morph.data[c("PC1_DorsalViewHead",
                                           "PC1_LateralViewHead")]),
            color = "grey85") +
  geom_point(color = "#3288BD", size = 2) +
  #stat_ellipse() +
  theme_bw() +
  facet_wrap(vars(Oz_Group))
```



```
ggplot(morph.data, aes(PC1_LinearMeasurements, PC1_LateralViewHead)) +
  geom_point(data = transform(morph.data[c("PC1_LinearMeasurements",
                                            "PC1_LateralViewHead")]),
            color = "grey85") +
  geom_point(color = "#D53E4F", size = 2) +
  #stat_ellipse() +
  theme_bw() +
  facet_wrap(vars(Oz_Group))
```



### Quick Take-away Thought:

From a preliminary view, it doesn't look like the three immigrant lineages (*doreanus*, *chlorostigma*, *keith-hornei*) are exceptional in any of the morphological dimensions. Maybe that makes sense as to why they haven't speciated in Australia? They really just make it in around the edges, and they aren't doing anything wildly novel, so they don't spread/speciate?

## Data Preparation

### Extracting branching times

We need to create an input file for DAISIE, consisting of the ages/names of the Australian clades. The radiation of subgenera *Varanus/Odatria* is the incumbent group, and there are three immigrant lineages (*V. keithhornei*, *V. doreanus*, *V. chlorostigma*).

We want to get the ages of these lineages in the Australian region.

```
ov.bt <- branching.times(extract.clade(vtree,
                                     getMRCA(vtree, c("komodoensis", "tristis"))))
names(ov.bt) <- NULL; ov.bt <- sort(ov.bt, decreasing=T)

vk.bt <- max(nodeHeights(vtree)) -
  nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "keithhornei")))
vd.bt <- max(nodeHeights(vtree)) -
  nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "doreanus")))
vc.bt <- max(nodeHeights(vtree)) -
  nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "chlorostigma")))
```

### Making a DAISIE datatable

We know the *Odatria/Varanus* radiation is endemic to Australia (minus a couple lineages that have since left - we need to think about how to handle these). *V. keithhornei* is a Cape York endemic species as well, but *V. doreanus* and *V. chlorostigma* exist elsewhere as well (New Guinea). We will characterize these latter two in our data as “Non\_endemic\_MaxAge”, because we know they aren’t endemic lineages, but we don’t know how old the Australian population is, though it must be younger than the age of their MRCAs.

```
DAISIE.oz <- data.frame(Clade_name = c("Odatria_Varanus",
                                       "V_keithhornei",
                                       "V_doreanus",
                                       "V_chlorostigma"),
                        Status = c("Endemic",
                                   "Endemic",
                                   "Non_endemic_MaxAge",
                                   "Non_endemic_MaxAge"),
                        Missing_species = c(0, 0, 0, 0),
                        Branching_times = c(paste(ov.bt, collapse=","),
                                             vk.bt, vd.bt, vc.bt))

DAISIE.oz[1:4,1:3]
```

##	Clade_name	Status	Missing_species
## 1	Odatria_Varanus	Endemic	0
## 2	V_keithhornei	Endemic	0
## 3	V_doreanus	Non_endemic_MaxAge	0
## 4	V_chlorostigma	Non_endemic_MaxAge	0

If you’d like, we can write this dataframe to a file which we can read in later.

```
write.table(DAISIE.oz, file = "DAISIE_oz_datatable.txt",
            sep = "\t", quote = F, row.names = F)
```



## Making a DAISIE datalist

Translate your data from a dataframe into a DAISIE datalist.

This first datalist assumes all taxa belong to a single grouping for the purposes of estimating parameters.

```
all.list <- DAISIE_dataprep(datatable = DAISIE.oz,  
                           island_age = 35, M = 30)
```

Create a second DAISIE datalist with two groupings (1:Odatria/Varanus, 2:all other immigrant groups). This will allow us to estimate separate parameter values for the two groups (if we'd like).

```
split.list <- DAISIE_dataprep(datatable = DAISIE.oz,  
                             island_age = 35, M = 30,  
                             number_clade_types = 2,  
                             list_type2_clades = c("V_keithornei",  
                                                    "V_doreanus",  
                                                    "V_chlorostigma"),  
                             prop_type2_pool = 0.1)
```

Check out the info that's held in the DAISIE datalist

```
# island info  
split.list[[1]]
```

```
## $island_age  
## [1] 35  
##  
## $not_present_type1  
## [1] 26  
##  
## $not_present_type2  
## [1] 0
```

```
# group 1 info  
split.list[[2]][1:2]
```

```
## $colonist_name  
## [1] "Odatria_Varanus"  
##  
## $branching_times  
## [1] 35.0000000 20.5629427 17.8867605 17.3072926 16.2756732 15.3659410  
## [7] 14.6671189 13.3416742 11.6340566 11.2317171 10.5024835 9.5688680  
## [13] 9.1319306 8.1492155 7.8125563 7.6881317 7.4196078 6.3215394  
## [19] 6.1738671 5.5353456 3.5395300 3.4929574 2.9702506 2.6626381  
## [25] 2.6449793 2.4431166 2.1974345 2.0568718 1.6463329 0.3817197  
## [31] 0.2797268
```

## Fitting DAISIE models

Below are examples of how to build and fit a number of models with DAISIE. I've left  $eval = F$  at the moment because fitting the model a single time is probably insufficient, as maximum likelihood can get trapped in local optima. We will come back to this in a minute.

Either way, fitting the models takes quite a bit of time, so my recommendation is to run them externally, then load the model fit objects and summarize/compare/plot them here to keep things clear.

---

### Equal Rates model with Diversity Dependence

5 parameters (cladogenetic speciation, extinction, carrying capacity, immigration rate, anagenetic speciation) are estimated across **ALL** taxa and groups.

```
eqr <- DAISIE_ML(datalist = all.list,
  initparsopt = c(0.5, 0.5, 30, 0.01, 0.1),
  idparsopt = 1:5,
  ddmodel = 11,
  parsfix = NULL,
  idparsfix = NULL)
```

---

### Equal Rates model without Diversity Dependence

4 parameters (cladogenetic speciation, extinction, immigration rate, anagenetic speciation) are estimated across **ALL** taxa and groups. There is no carrying capacity (param 3), so it is fixed at *Infinity*.

```
eqr_noDD <- DAISIE_ML(datalist = all.list,
  initparsopt = c(0.5, 0.5, 0.01, 0.1),
  idparsopt = c(1,2,4,5),
  ddmodel = 11,
  parsfix = Inf,
  idparsfix = 3)
```

---

### Two Rate Speciation model with Diversity Dependence

4 parameters (extinction, carrying capacity, immigration rate, anagenetic speciation) are estimated across **ALL** taxa and groups, but we estimate separate cladogenetic speciation rates for the two groups (*Odatria*/*Varanus* and immigrants).

```
two_lambda <- DAISIE_ML(datalist = split.list,
  ddmodel = 11,
  initparsopt = c(1, 0.1, 30, 0.005, 0.1,
    1),
  idparsopt = c(1,2,3,4,5,6),
  parsfix = NULL,
  idparsfix = NULL,
  idparsnoshift = c(7,8,9,10))
```

## Two Rate Speciation and Extinction model with Diversity Dependence

3 parameters (carrying capacity, immigration rate, anagenetic speciation) are estimated across **ALL** taxa and groups, but we estimate separate cladogenetic speciation and extinction rates for the two groups (*Oda-tria*/*Varanus* and immigrants).

```
two_lambda_mu <- DAISIE_ML(datalist = split.list,
                           ddmodel = 11,
                           initparsopt = c(1, 0.1, 30, 0.005, 0.1,
                                             1, 0.1),
                           idparsopt = c(1,2,3,4,5,6,7),
                           parsfix = NULL,
                           idparsfix = NULL,
                           idparsnoshift = c(8,9,10))
```

---

## Comparing the fit of DAISIE Models

Compare the fit of the models using the custom *DAISIE.AIC* function.

```
source("../Scripts/Calculate_AICs.R")
DAISIE.AIC(models = c("eqr", "two_lambda"))
```

## Fitting DAISIE Models in Parallel

Import a custom function to fit DAISIE models multiply from different starting parameter values

```
source("../Scripts/search.surface.DAISIE.R")
```

I'll try and quickly explain what you need to provide the function:

- **model**—the DAISIE datalist suited for the model you'd like to fit
- **n.iter**—the number of model fittings you'd like completed, defaults to 10
- **n.proc**—number of processors. this function will fit the model in parallel
- **g1.params**—if you'd like to provide starting values for parameters 1:5 (`lambda_c`, `mu`, `K`, `gamma`, `lambda_a`) do that here. Otherwise, they'll be drawn randomly from a reasonable (flat) prior
- **g2.params**—if you need to estimate parameters for a second group 6:10 (`lambda_c`, `mu`, `K`, `gamma`, `lambda_a`) do that here. They'll be drawn randomly from a reasonable (flat) prior. I could design it to work like `g1.params`, but I just haven't yet
- **results**—would you like the function to report just the best fitting run, or all the results from each fit attempt
- **start.params**—ignore this, just leave it as `NULL`
- **dd.model**—same as `ddmodel` in DAISIE, 11 for linear density dependence, etc.
- **id.parsopt**—same as `idparsopt` in DAISIE, ids for which parameters to optimize
- **pars.fix**—same as `parsfix` in DAISIE, which parameters to fix
- **id.parsfix**—same as `idparsfix` in DAISIE, ids for the parameters to fix

---

## Equal Rates model with Diversity Dependence

5 parameters (cladogenetic speciation, extinction, carrying capacity, immigration rate, anagenetic speciation) are estimated across **ALL** taxa and groups.

```
eqr <- search.surface.DAISIE(model = all.list, n.iter = 10, n.proc = 12,  
                             g1.params = NULL, g2.params = NULL,  
                             results = "all", start.params = NULL,  
                             dd.model = 11,  
                             id.parsopt = c(1:5),  
                             pars.fix = NULL,  
                             id.parsfix = NULL)
```

If you fit the model with `results = "all"`, then you'll get all the model fittings, plus a designated *best.result* which we can look at below:

```
eqr$best.result
```

Otherwise, if you fit `results = "best"`, then you only return the best fitting version.

---

## Two Speciation Rate model with Diversity Dependence

4 parameters (extinction, carrying capacity, immigration rate, anagenetic speciation) are estimated across **ALL** taxa and groups, but we estimate separate cladogenetic speciation rates for the two groups (*Odatria*/*Varanus* and immigrants).

```
two_lambda <- search.surface.DAISIE(model = split.list, n.iter = 3, n.proc = 3,  
                                    g1.params = NULL, g2.params = c("lambda_c"),  
                                    results = "all", start.params = NULL,
```

```
dd.model = 11,  
id.parsopt = c(1:6),  
pars.fix = NULL,  
id.parsfix = NULL,  
id.parsnoshift = c(7:10))
```

---

I'll leave a gap here to build more models or show how we could construct them at least.

---

## Comparing the fit of DAISIE Models run in parallel

Compare the fit of the models using the custom *DAISIE.AIC* function.

```
source("../Scripts/Calculate_AICs.R")  
equal_rates <- eqr$best.result  
two_sp <- two_lambda$best.result  
DAISIE.AIC(models = c("equal_rates", "two_sp"))
```

# Diversification Models with ClaDS

Direct your working directory to the folder that holds the ClaDS scripts

```
setwd("~/Google.Drive/R.Analyses/ClaDS/")
```

Load RPANDA

```
library(RPANDA)
```

In case you're jumping straight to here, start by reading in the tree

```
vtree <- read.tree("UsedTree.tree")
```

ClaDS (within RPANDA) has a couple of models we can fit • **ClaDS0**—assumes pure birth process when estimating speciation rates (extinction = 0)

- **ClaDS1**—assumes a constant extinction rate, and estimates variable speciation rates
- **ClaDS2**—estimates variable speciation and extinction rates, but holds turnover rates constant

Fitting these models takes a lot of time, and generations, so make sure to set the iterations relatively high. You can extend runs if you haven't reached convergence though.

---

## ClaDS0—Pure Birth Model

```
clads0.varanus <- fit_ClaDS0(tree = vtree,  
                             sample_fraction = (76/83),  
                             iterations = 10000,  
                             thin = 100,  
                             pamhLocalName = "local",  
                             #it_save = 10000,  
                             name = "Monitor_ClaDS0",  
                             nCPU = 3)
```

We can check the chains for convergence

```
plot_ClaDS_chains(clads2.varanus)
```

Then we can extract the per branch rates, and plot them along the tree

```
maps <- getMAPS_ClaDS(clads2.varanus, thin = 10)  
plot_ClaDS_phylo(vtree, maps[-(1:4)])
```

---

## ClaDS2—Constant Turnover Model

```
clads2.varanus <- fit_ClaDS(tree = vtree,  
                             sample_fraction = (76/83),  
                             iterations = 10000,  
                             thin = 100,  
                             #it_save = 10000,  
                             file_name = "Monitor_ClaDS2",
```

```
model_id = "ClaDS2",
nCPU = 3)
```

We can check the chains for convergence

```
plot_ClaDS_chains(clads2.varanus)
```

Then we can extract the per branch rates and plot them along the tree

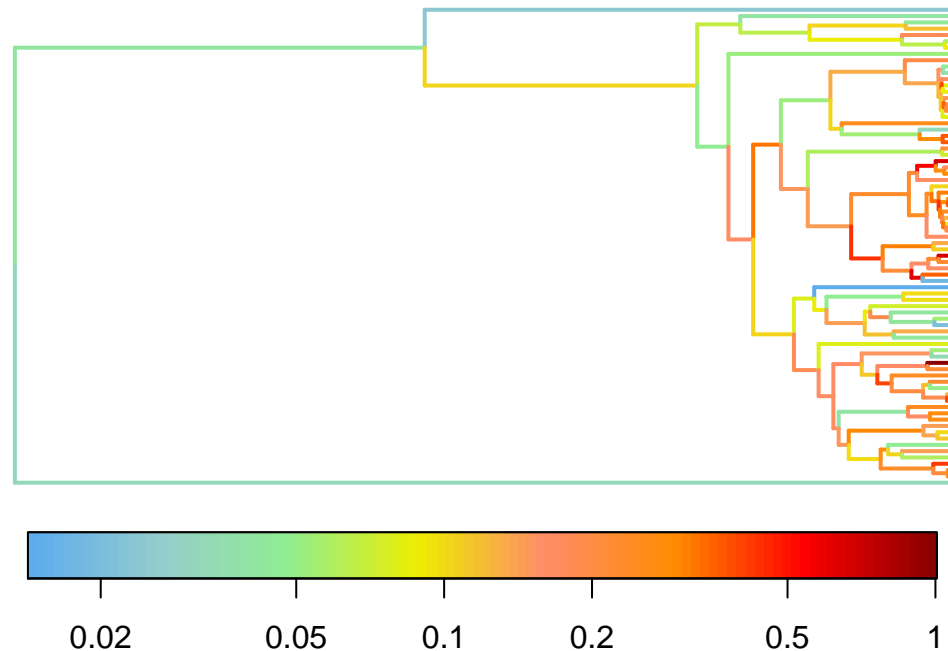
```
maps <- getMAPS_ClaDS(clads2.varanus, thin = 10)
plot_ClaDS_phylo(vtree, maps[-(1:4)])
```

Here's an example file I didn't run long enough:

```
c2_run1 <- readRDS("~/Documents/GitHub/MonitorDiversification/Data/ClaDS2_Varanus.RDS")
plot_ClaDS_chains(c2_run1)
```

Let's look at the plot anyway, then make sure to run for more generations

```
maps <- getMAPS_ClaDS(c2_run1, thin = 10)
plot_ClaDS_phylo(vtree, maps[-(1:4)])
```

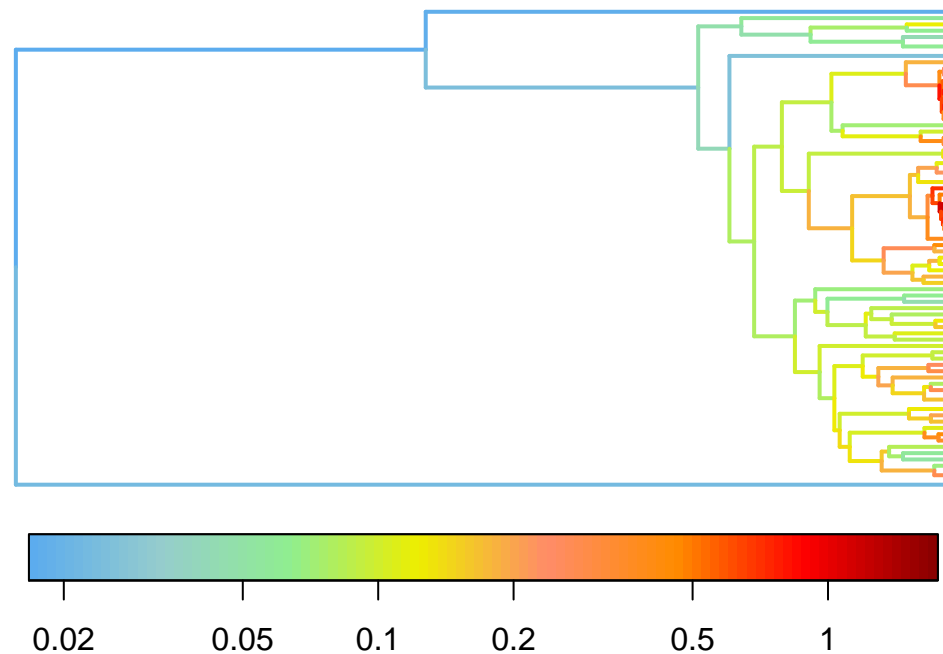


The above example was run for 10,000 generations, thin = 100. Run it longer by just providing the previous file

```
c2_run1_cont <- fit_ClaDS(tree = vtree,
  sample_fraction = (76/83),
  iterations = 10000,
  model_id = "ClaDS2",
  nCPU = 3,
  sampler = c2_run1)
```

We can read in a longer pre-cooked file. This one has been run for 200K generations and 3 chains.

```
load("~/Documents/GitHub/MonitorDiversification/Data/C2_Varanus_200K.RData")
c2.varanus <- mcmcSampler
maps <- getMAPS_ClaDS(c2.varanus, thin = 10)
plot_ClaDS_phylo(vtree, maps[-(1:4)])
```



We might want to compare the speciation rates across different subgenera, or clades. Start by making a function to extract the edge (branch) specific rates from the ClaDS object.

```
extract.ClaDS.rates <- function(rates, min.branch, max.branch){
  clade.rates <- rates[(min.branch + 4):(max.branch + 4)]
  return(clade.rates)}
```

This function is just really simple, and excludes the first four parameters (sigma, alpha, epsilon, l\_0), when isolating the rates of interest.



Now pick a handful of clades to compare:

```
odatria.rates <- extract.ClaDS.rates(maps, 7, 49)
hapturo.rates <- extract.ClaDS.rates(maps, 69, 83)
euprep.rates <- extract.ClaDS.rates(maps, 83, 101)
varanus.rates <- extract.ClaDS.rates(maps, 50, 66)
sotero.rates <- extract.ClaDS.rates(maps, 119, 137)
empagus.rates <- extract.ClaDS.rates(maps, 112, 118)
african.rates <- extract.ClaDS.rates(maps, 139, 149)
```

Put the rates together into a single data frame

```
all.rates <- data.frame(sp_rates = odatria.rates, group = "Odatria")
all.rates <- add_row(all.rates,
                    sp_rates = hapturo.rates,
                    group = "Hapturosaurus")
all.rates <- add_row(all.rates,
                    sp_rates = euprep.rates,
                    group = "Euprepiosaurus")
all.rates <- add_row(all.rates,
                    sp_rates = varanus.rates,
                    group = "Varanus_Papusaurus")
all.rates <- add_row(all.rates,
                    sp_rates = sotero.rates,
                    group = "Soterosaurus")
all.rates <- add_row(all.rates,
                    sp_rates = empagus.rates,
                    group = "Empagusia")
all.rates <- add_row(all.rates,
                    sp_rates = african.rates,
                    group = "Polydaedalus_Psammosaurus")
```

And plot these to get an idea of variation across the groups:

```
library(ggribes)
ggplot(all.rates, aes(x=sp_rates, y=group,
                     point_color=sp_rates)) +
  geom_density_ridges(jittered_points = T, scale = 2,
                    point_shape = "|", alpha = 0.75,
                    point_size = 2, size = 0.5,
                    position = position_points_jitter(height=0),
                    aes(fill = group), show.legend = F) +
  scale_fill_brewer(palette = "RdYlBu") +
  theme_bw()
```

