Monitor Lizard Diversification with DAISIE

Ian G. Brennan & Carlos Pavon Vazquez

09 September, 2020

Contents

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 is currently set to **private**).

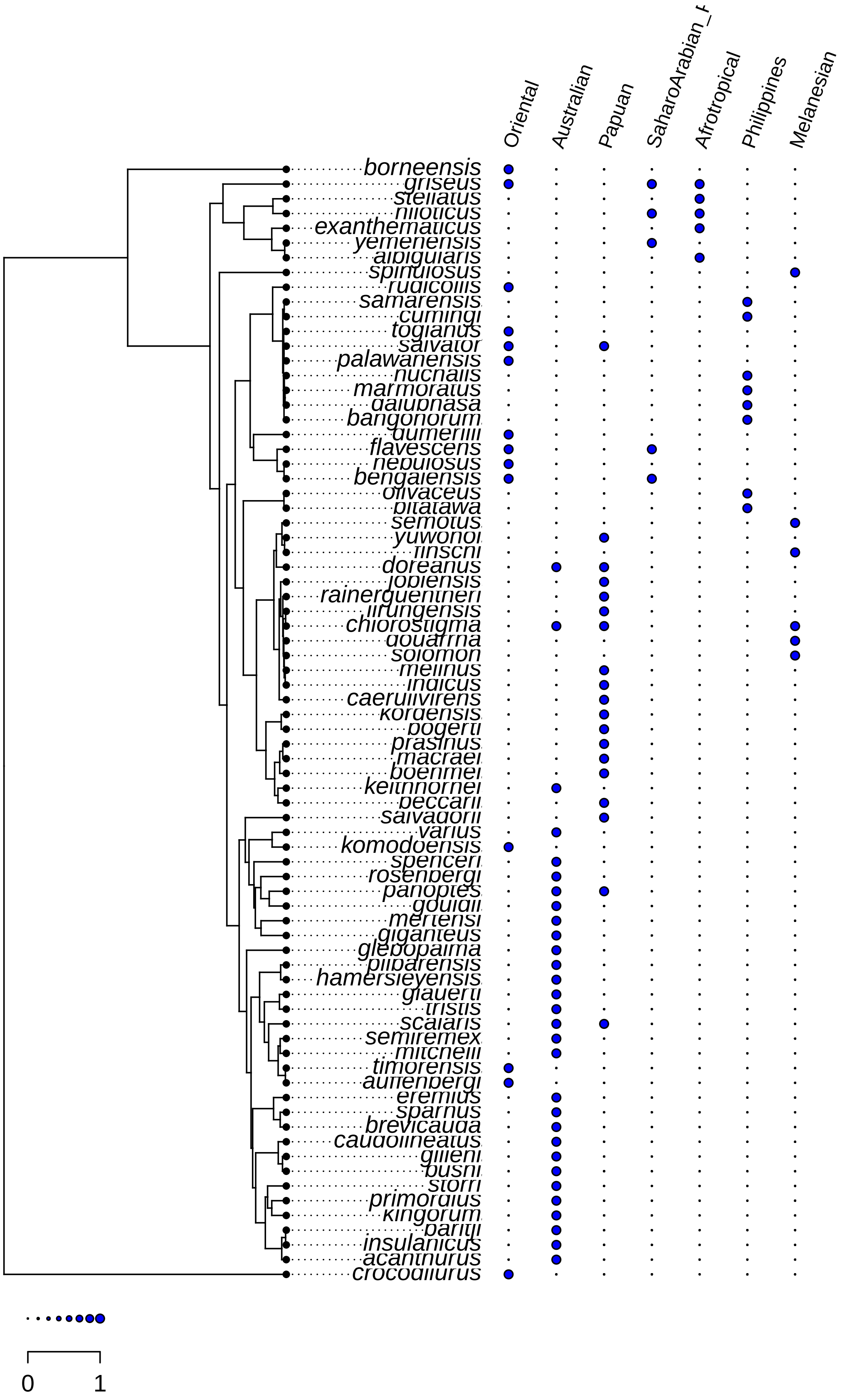
Start off by loading a few packages that we'll need along the way.

```r
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

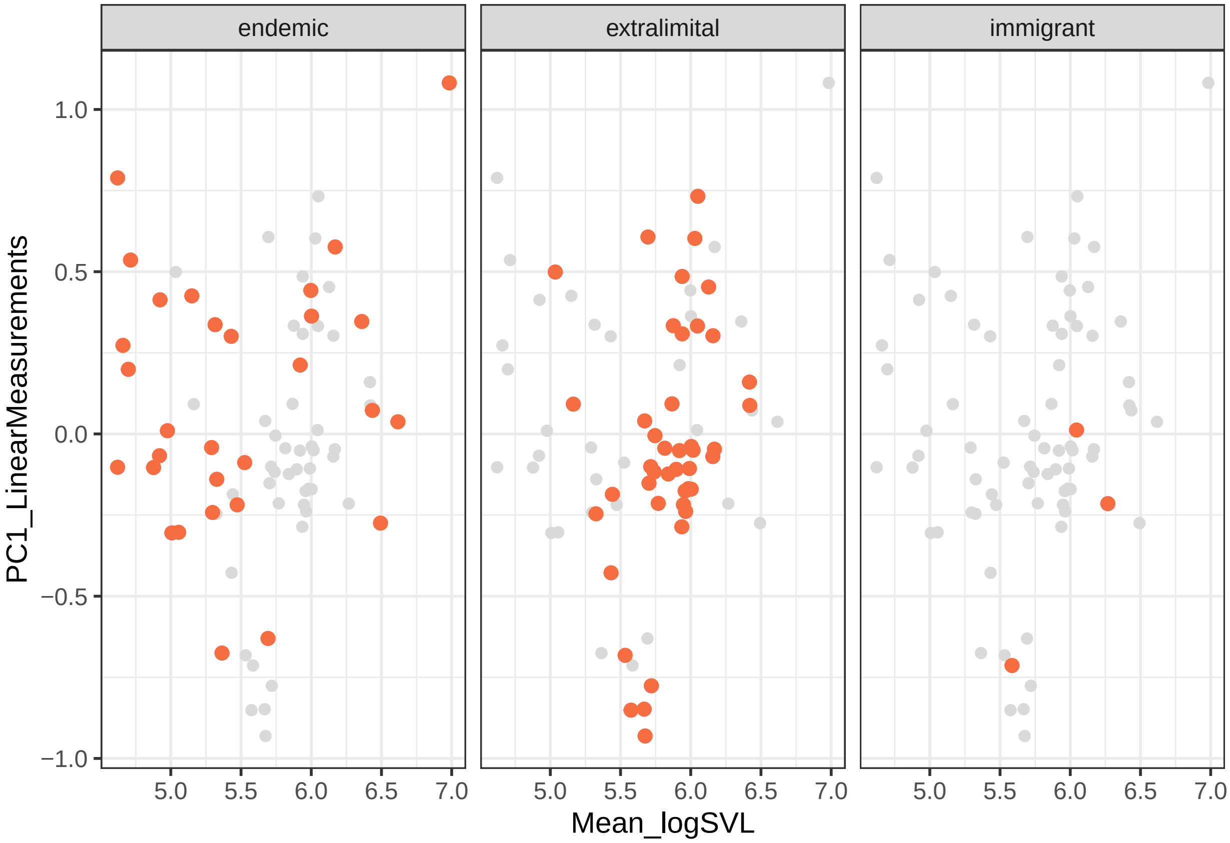
```r
vtree <- read.tree("UsedTree.tre");
regions <- read.delim("geo_file_Regions.txt", sep="\t", row.names = 1)
dotTree(vtree, regions, labels=T,length=8)
```


Oriental
Australian
Papuan
SaharoArabian_
Afrotropical
Philippines
Melanesian
borneensis
griseus
stellatus
niloticus
exanthematicus
yemenensis
albigularis
spinulosus
rudicollis
samarensis
cumingi
togianus
salvator
palawanensis
nuchalis
marmoratus
dalubhasa
bangonorum
dumerilii
flavescens
nebulosus
bengalensis
olivaceus
bitatawa
semotus
yuwonoi
finschi
doreanus
jobiensis
rainerguentheri
lirungensis
chlorostigma
douarrha
solomon
melinus
indicus
caerulivirens
kordensis
bogerti
prasinus
macraei
boehmei
keithhornei
beccarii
salvadorii
varius
komodoensis
spenceri
rosenbergi
panoptes
gouldii
mertensi
giganteus
glebopalma
pilbarensis
hamersleyensis
glauerti
tristis
scalaris
semiremex
mitchelli
timorensis
auffenbergi
eremius
sparnus
brevicauda
caudolineatus
gilleni
bushi
storri
primordius
kingorum
baritji
insulanicus
acanthurus
crocodilurus
0
1

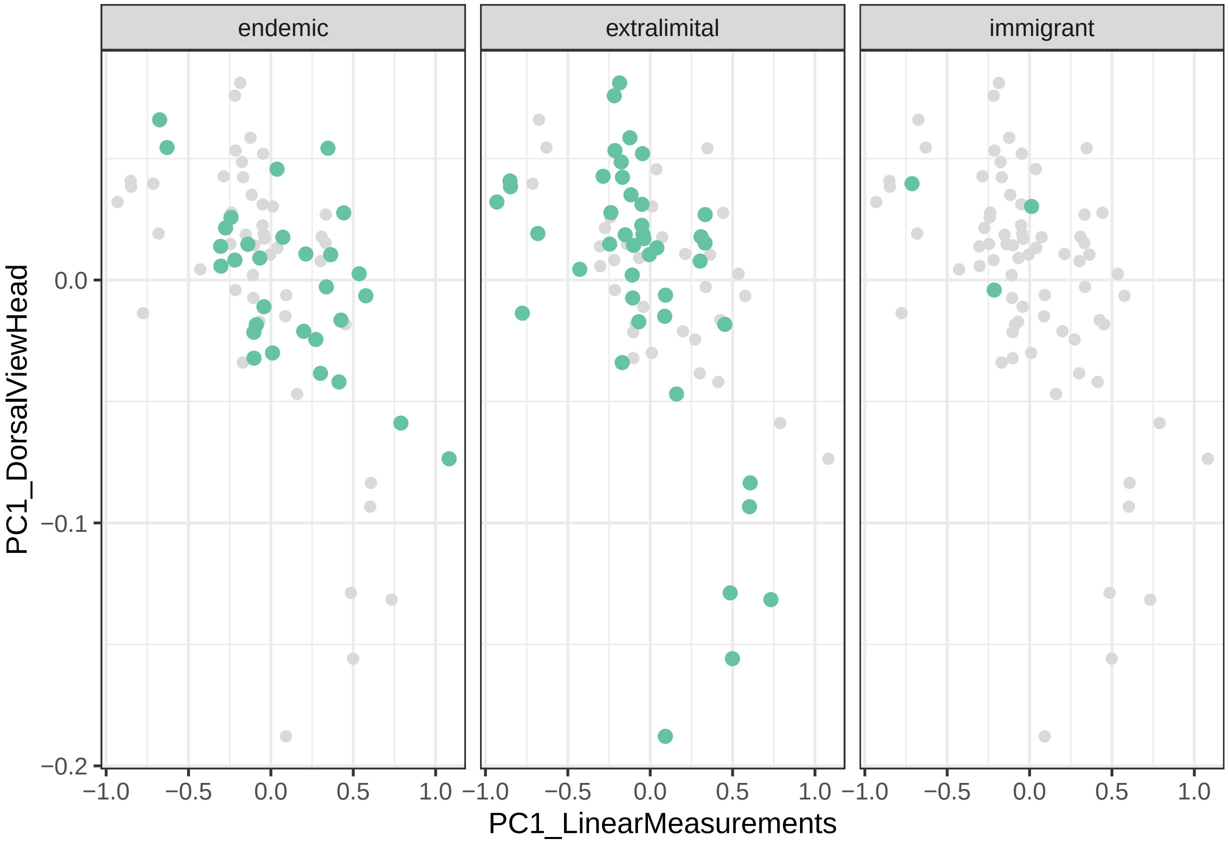
Read in the morphological data we have and visualize data.
I've made an additional data column to look at the Australian monitor data. I've categorized them as "endemic" to Australia (most of the *Odatria/Varanus* subgenera), "immigrant" (*doreanus/chlorostigma/keithhornei*), or "extralimital" which is just all things existing elsewhere exclusively.

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

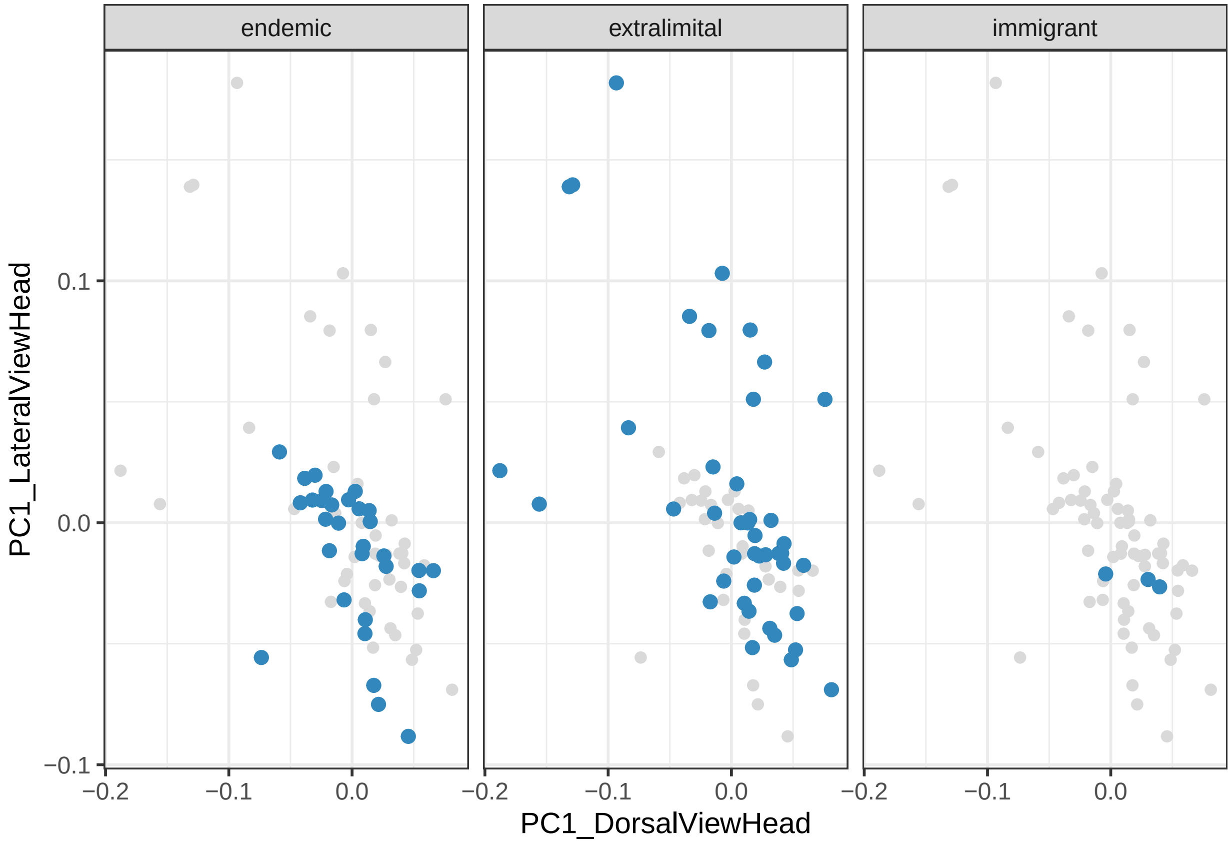
```
ggplot(morph.data, aes(Mean_logSVL, PC1_LinearMeasurements)) +
  geom_point(data = transform(morph.data[c("Mean_logSVL",
                                           "PC1_LinearMeasurements")]),
             color = "grey85") +
  geom_point(color = "#F46D43", size = 2) +
  #stat_ellipse() +
  theme_bw()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



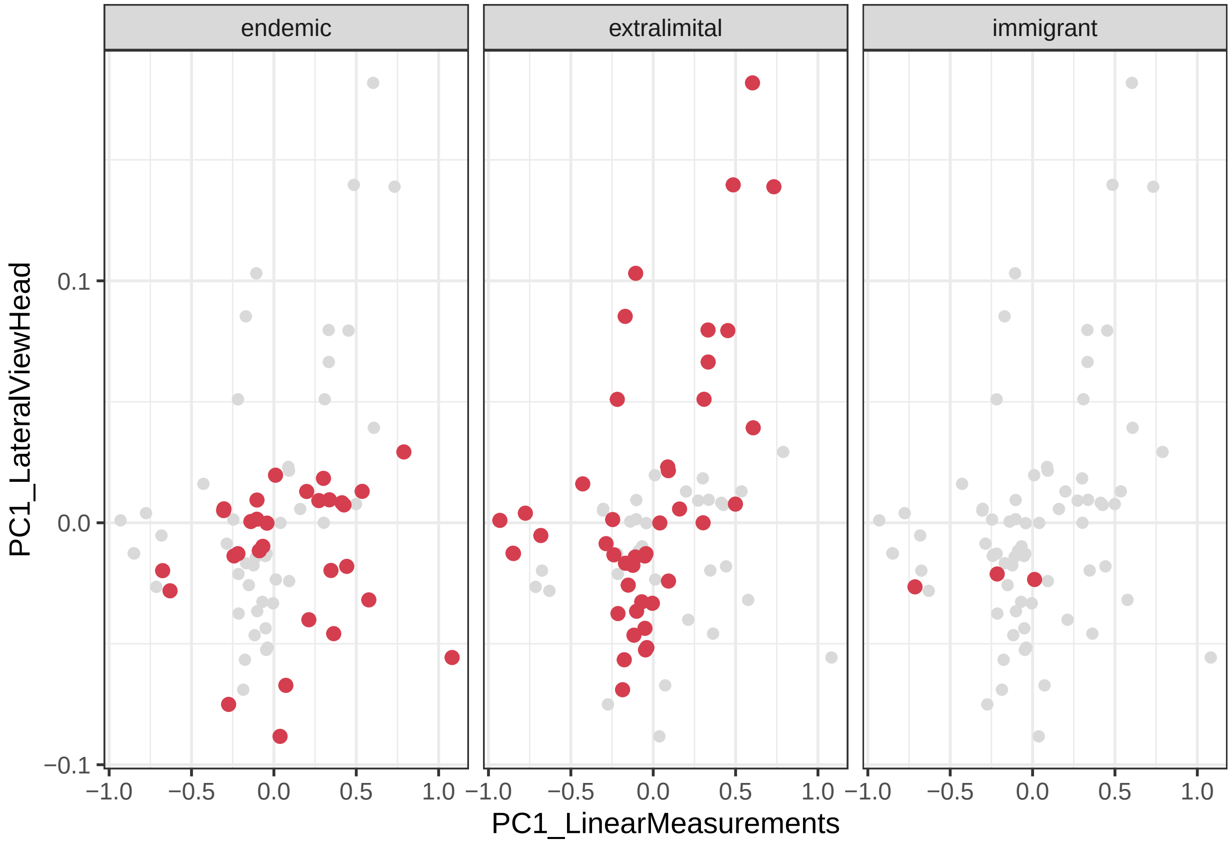
```
ggplot(morph.data, aes(PC1_LinearMeasurements, PC1_DorsalViewHead)) +
  geom_point(data = transform(morph.data[c("PC1_LinearMeasurements",
                                           "PC1_DorsalViewHead")]),
             color = "grey85") +
  geom_point(color = "#66C2A5", size = 2) +
  #stat_ellipse()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



```r
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, keithhornei*) 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 incumbent clades and immigrant groups.

If you have questions about implementing the DAISIE models, we can check with the package tutorial:

```
DAISIE_tutorial()
```

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")))
```

Let's do the same thing for the Papuan radiation

```
pa.bt <- branching.times(extract.clade(vtree,
                                       getMRCA(vtree, c("indicus", "prasinus"))))
names(pa.bt) <- NULL; pa.bt <- sort(pa.bt, decreasing=T)

vsa.bt <- max(nodeHeights(vtree)) -
         nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "salvadorii")))
vsc.bt <- max(nodeHeights(vtree)) -
         nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "scalaris")))
vst.bt <- max(nodeHeights(vtree)) -
         nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "salvator")))
vpa.bt <- max(nodeHeights(vtree)) -
         nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "panoptes")))
```

And again for the Asian mainland group

```
as.bt <- branching.times(extract.clade(vtree,
                                       getMRCA(vtree, c("dumerilii", "bengalensis"))))
names(as.bt) <- NULL; as.bt <- sort(as.bt, decreasing=T)

vr.bt <- max(nodeHeights(vtree)) -
         nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "rudicollis")))
vst.bt <- max(nodeHeights(vtree)) -
         nodeheight(vtree, getParent(vtree, which(vtree$tip.label == "salvator")))
```

Once more for the combined Incumbent vs. Immigrant scenario

```
inc.bt <- c(ov.bt, pa.bt, as.bt)
```

Making a DAISIE datatable

We know the *Odatria/Varanus* radiation is endemic to Australia . *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.

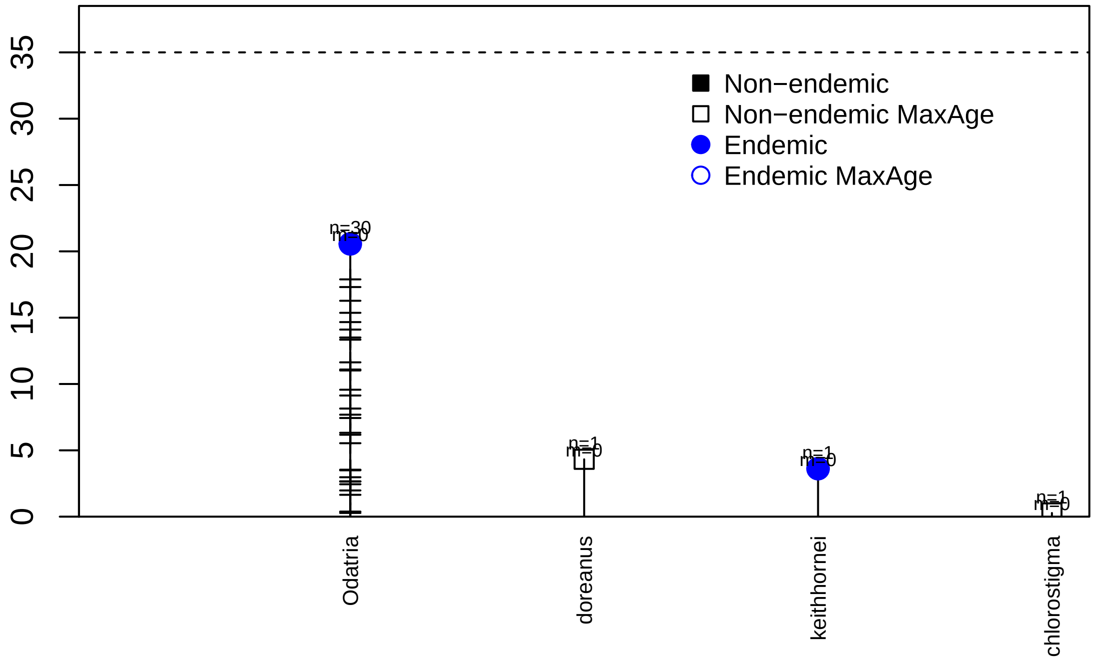
```
DAISIE.oz <- data.frame(Clade_name = c("Odatria_Varanus",
                                       "keithhornei",
                                       "doreanus",
                                       "chlorostigma"),
                        Status = c("Endemic",
                                   "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     keithhornei            Endemic               0
## 3        doreanus Non_endemic_MaxAge               0
## 4    chlorostigma Non_endemic_MaxAge               0
```

```
Australia.endemics <- data.frame(Clade_name = "Odatria_Varanus",
                                 Status = "Endemic",
                                 Missing_species = 0,
                                 Branching_times = c(paste(ov.bt, collapse=",")))
Australia.immigrants <- data.frame(Clade_name = c("keithhornei", "doreanus", "chlorostigma"),
                                   Status = c("Endemic", "Non_endemic_MaxAge", "Non_endemic_MaxAge"),
                                   Missing_species = c(0, 0, 0),
                                   Branching_times = c(vk.bt, vd.bt, vc.bt))
```

View what the distribution of taxa ages and status looks like

```
DAISIE_plot_island(DAISIE.oz, island_age = 35)
```



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)
```

Let's do the same for the *Papuan* radiation. The incumbent (endemic) radiation is Hapturosaurus + Euprepiosaurus (mangrove and tree monitors), and then *V. salvadorii*, *V. scalaris*, and *V. panoptes* have emigrated from Australia, and *V.salvator* has emigrated from Indonesia. All three of the four species exist elsewhere and we don't know how long they've been in New Guinea, but the fourth *V. salvadorii* is an endemic lineage.

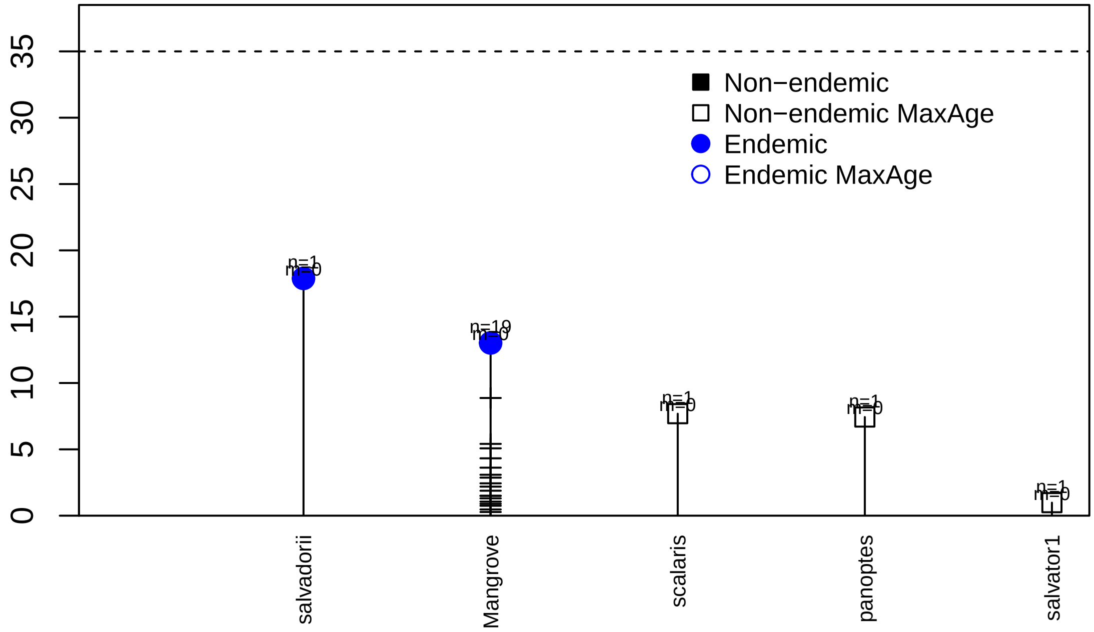
```
DAISIE.pa <- data.frame(Clade_name = c("Mangrove_Tree",
                                       "salvadorii",
                                       "scalaris",
                                       "salvator1",
                                       "panoptes"),
                        Status = c("Endemic",
                                   "Endemic",
                                   "Non_endemic_MaxAge",
                                   "Non_endemic_MaxAge",
                                   "Non_endemic_MaxAge"),
                        Missing_species = c(0, 0, 0, 0, 0),
                        Branching_times = c(paste(pa.bt, collapse=","),
                                            vsa.bt, vsc.bt, vst.bt, vpa.bt))
DAISIE.pa[1:5,1:3]
```

```
##      Clade_name             Status Missing_species
## 1 Mangrove_Tree            Endemic               0
## 2    salvadorii            Endemic               0
## 3      scalaris Non_endemic_MaxAge               0
## 4     salvator1 Non_endemic_MaxAge               0
## 5      panoptes Non_endemic_MaxAge               0
```

```
NewGuinea.endemics <- data.frame(Clade_name = "Mangrove_Tree",
                                 Status = "Endemic",
                                 Missing_species = 0,
                                 Branching_times = c(paste(pa.bt, collapse=",")))
NewGuinea.immigrants <- data.frame(Clade_name = c("salvadorii", "scalaris", "salvator1", "panoptes"),
                                   Status = c("Endemic", "Non_endemic_MaxAge", "Non_endemic_MaxAge", "N
                                   Missing_species = c(0, 0, 0, 0),
                                   Branching_times = c(vsa.bt, vsc.bt, vst.bt, vpa.bt))
```

View what the distribution of taxa ages and status looks like

```
DAISIE_plot_island(DAISIE.pa, island_age = 35)
```



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

```
write.table(DAISIE.pa, file = "Data/DAISIE_pa_datatable.txt",
            sep = "\t", quote = F, row.names = F)
```

And one more time for the mainland Asian radiation. The incumbent (endemic) radiation is Empagusia, and *V. rudicollis* and *V. salvator* have arrived from the east. *V. rudicollis* is endemic, and like in the Papuan situation, *V. salvator* is an immigrant and we don't know how long they've been around for.

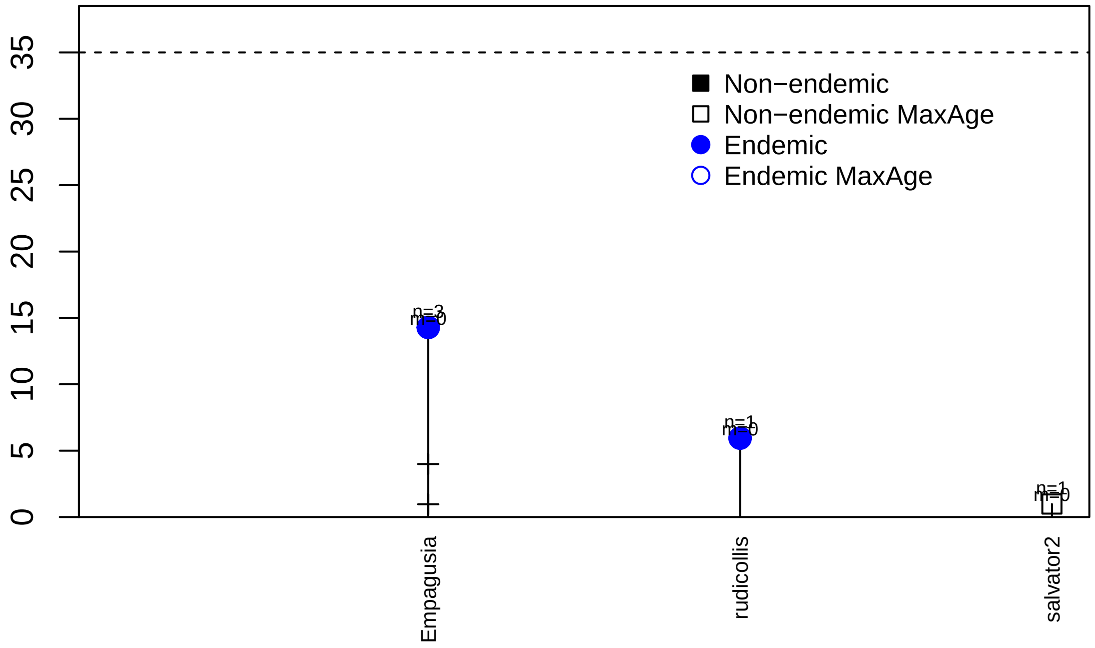
```
DAISIE.as <- data.frame(Clade_name = c("Empagusia",
                                       "rudicollis",
                                       "salvator2"),
                        Status = c("Endemic",
                                   "Endemic",
                                   "Non_endemic_MaxAge"),
                        Missing_species = c(0, 0, 0),
                        Branching_times = c(paste(as.bt, collapse=","),
                                            vr.bt, vst.bt))
DAISIE.as[1:3,1:3]
```

```
##   Clade_name             Status Missing_species
## 1  Empagusia            Endemic               0
## 2 rudicollis            Endemic               0
## 3  salvator2 Non_endemic_MaxAge               0
```

```
Asia.endemics <- data.frame(Clade_name = "Empagusia",
                                 Status = "Endemic",
                                 Missing_species = 0,
                                 Branching_times = c(paste(as.bt, collapse=",")))
Asia.immigrants <- data.frame(Clade_name = c("rudicollis", "salvator2"),
                                   Status = c("Endemic", "Non_endemic_MaxAge"),
                                   Missing_species = c(0, 0),
                                   Branching_times = c(vr.bt, vst.bt))
```

View what the distribution of taxa ages and status looks like

```
DAISIE_plot_island(DAISIE.as, island_age = 35)
```



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

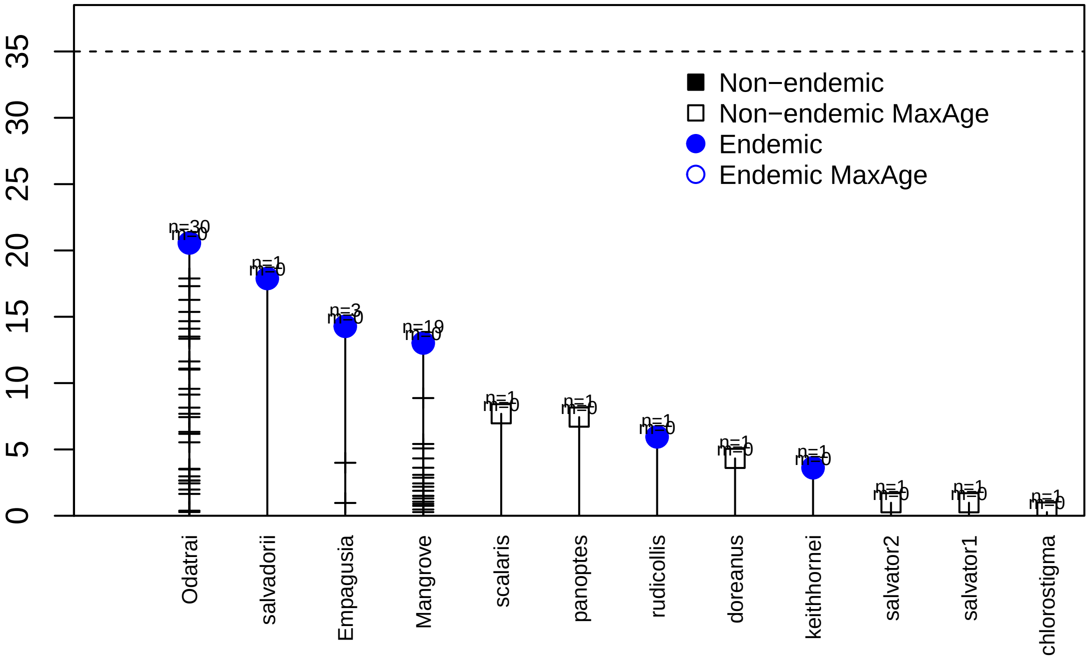
```
write.table(DAISIE.as, file = "DAISIE_as_datatable.txt",
            sep = "\t", quote = F, row.names = F)
```

Let's do this one last time grouping all the incumbent groups together (Empagusia, Odatria/Varanus, Hapturosaurus/Euprepiosaurus), and all the immigrants together, to see if we can find stronger signal in this combined dataset.

```
DAISIE.combo <- data.frame(Clade_name = c("Odatrai_Varanus",
                                          "keithhornei",
                                          "doreanus",
                                          "chlorostigma",
                                          "Mangrove_Tree",
                                          "salvadorii",
                                          "scalaris",
                                          "salvator1",
                                          "panoptes",
                                          "Empagusia",
                                          "rudicollis",
                                          "salvator2"),
                           Status = c("Endemic",
                                      "Endemic",
                                      "Non_endemic_MaxAge",
                                      "Non_endemic_MaxAge",
                                      "Endemic",
                                      "Endemic",
                                      "Non_endemic_MaxAge",
                                      "Non_endemic_MaxAge",
                                      "Non_endemic_MaxAge",
                                      "Endemic",
                                      "Endemic",
                                      "Non_endemic_MaxAge"),
                           Missing_species = c(0,0,0,0,0,0,0,0,0,0,0,0),
                           Branching_times = c(paste(ov.bt, collapse=","), vk.bt, vd.bt, vc.bt,
                                               paste(pa.bt, collapse=","), vsa.bt, vsc.bt, vst.bt, vpa.b
                                               paste(as.bt, collapse=","), vr.bt, vst.bt))
```

View what the distribution of taxa ages and status looks like

```
DAISIE_plot_island(DAISIE.combo, island_age = 35)
```



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. We'll do this for each of the areas we're investigating. This process also requires that we designate an island

age, and the number of lineages that could be in the "source" population (really just number of total number of species - endemics, I think).

```
all.list.oz    <- DAISIE_dataprep(datatable = DAISIE.oz,
                              island_age = 35, M = 30)
#names(all.list.oz) <- rep("Australia", length(all.list.oz))

all.list.pa    <- DAISIE_dataprep(datatable = DAISIE.pa,
                              island_age = 35, M = 50)
#names(all.list.pa) <- rep("NewGuinea", length(all.list.pa))

all.list.as    <- DAISIE_dataprep(datatable = DAISIE.as,
                              island_age = 35, M = 68)
#names(all.list.as) <- rep("Asia", length(all.list.as))

all.list.combo <- list(Australia = all.list.oz,
                       NewGuinea = all.list.pa,
                       Asia = all.list.as)

AUin.list <- DAISIE_dataprep(datatable = Australia.endemics,
                             island_age = 35, M = 30)
AUim.list <- DAISIE_dataprep(datatable = Australia.immigrants,
                             island_age = 35, M = 30)
NGin.list <- DAISIE_dataprep(datatable = NewGuinea.endemics,
                             island_age = 35, M = 50)
NGim.list <- DAISIE_dataprep(datatable = NewGuinea.immigrants,
                             island_age = 35, M = 50)
ASin.list <- DAISIE_dataprep(datatable = Asia.endemics,
                             island_age = 35, M = 68)
ASim.list <- DAISIE_dataprep(datatable = Asia.immigrants,
                             island_age = 35, M = 68)

combined.list <- list(AUin = AUin.list, AUim = AUim.list,
                      NGin = NGin.list, NGim = NGim.list,
                      ASin = ASin.list, ASim = ASim.list)
```

Create a second DAISIE datalist with two groupings (1:Odatria/Varanus, 2:all other immigrant groups; 1:Euprepiosaurus/Hapturosaurus, 2:all immigrants; 1:Empagusia, 2:all immigrants). This will allow us to estimate separate parameter values for the two groups (incumbent vs. immigrant).
Here we also have to designate the proportion of the source species that the immigrants are (n-immigrants / n-source), which is *prop_type2_pool*.

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

split.list.pa <- DAISIE_dataprep(datatable = DAISIE.pa,
                              island_age = 35, M = 50,
                              number_clade_types = 2,
                              list_type2_clades = c("salvadorii",
                                                    "scalaris",
                                                    "salvator1",
                                                    "panoptes"),
                              prop_type2_pool = 0.08)

split.list.as <- DAISIE_dataprep(datatable = DAISIE.as,
                              island_age = 35, M = 68,
                              number_clade_types = 2,
                              list_type2_clades = c("rudicollis",
                                                    "salvator2"),
                              prop_type2_pool = 0.03)

#split.list.ic <- DAISIE_dataprep(datatable = DAISIE.combo,
#                              island_age = 35, M = 75,
#                              number_clade_types = 2,
#                              list_type2_clades = c("keithhornei", "doreanus", "chlorostigma",
#                                                    "salvadorii", "scalaris", "salvator1", "panoptes",
#                                                    "rudicollis", "salvator2"),
#                              prop_type2_pool = 0.1)

#split.list.ic <- list(Australia = split.list.oz,
#                      NewGuinea = split.list.pa,
#                      Asia = split.list.as)
```

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

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

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

```
# group 1 info
split.list.oz[[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 14.1008829 13.5052185 13.3416742 11.6340566 11.0896892
## [13] 11.0197803  9.5688680  9.1319306  8.1492155  7.6881317  7.4362806
## [19]  6.3215394  6.1738671  5.5353456  3.5395300  3.4929574  2.9702506
## [25]  2.6626381  2.6449793  2.4431166  1.9755136  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 (sub)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 clea

---

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_0 <- DAISIE_ML(datalist = all.list.oz,
                 initparsopt = c(0.2, 0.15, 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_0 <- DAISIE_ML(datalist = all.list.oz,
                      initparsopt = c(0.2, 0.15, 0.01, 0.01),
                      idparsopt = c(1,2,4,5),
                      parsfix = Inf,
                      ddmodel = NULL,
                      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.oz,
                        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 (*Odatria/Varanus* and immigrants).

```
two_lambda_mu <- DAISIE_ML(datalist = split.list.oz,
                           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 = 0.8,
                           idparsfix = 11,
                           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"))
#DAISIE.AIC(models = c("eqr", "eqr_noDD"))
```

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
- **g1.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.oz, 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,
                             id.parsnoshift = NULL,
                             no.types = 1)
```

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.

Before we get going, let's designate the group we want to work with

```
curr.group <- all.list.pa
curr.split <- split.list.pa
ntaxa <- 23
samp.prop <- 0.9
```

Pure Birth Models

In all Pure Birth models, the extinction rate is 0, so this parameter is not estimated (fixed).

Pure-Birth without Diversity Dependence

This model estimates a single speciation rate shared by the two groups, extinction is set to zero, and there is no diversity dependence (carrying capacity K = Infinite).

```
eqpb_noDD <- search.surface.DAISIE(model = curr.group, n.iter = 10, n.proc = 12,
                              g1.params = NULL, g2.params = NULL,
                              results = "all",
                              start.params = NULL,
                              dd.model = 0,
                              id.parsopt = c(1,4,5),
                              pars.fix = c(0,Inf),
                              id.parsfix = c(2,3),
                              no.types = 1,
                              nsamples = ntaxa)
```

PB with Speciation Linear to DD

Pure-Birth model with equal speciation rates linear to Diversity Dependence This model estimates a single speciation rate shared by the two groups, extinction is set to zero, and the speciation rate is linearly dependent on the carrying capacity K (linear diversity dependence, ddmodel = 1).

```
eqpb_linDD_1 <- search.surface.DAISIE(model = curr.group, n.iter = 10, n.proc = 12,
                              g1.params = NULL, g2.params = NULL,
                              results = "all",
                              start.params = NULL,
                              dd.model = 1,
                              id.parsopt = c(1,3,4,5),
                              pars.fix = 0,
                              id.parsfix = 2,
                              no.types = 1,
                              nsamples = ntaxa)
```

PB with Speciation Exponential to DD

Pure-Birth model with equal speciation rates exponential to Diversity Dependence This model estimates a single speciation rate shared by the two groups, extinction is set to zero, and the speciation rate is exponentially dependent on the carrying capacity K (exponential diversity dependence, ddmodel = 2).

```
eqpb_expDD_2 <- search.surface.DAISIE(model = curr.group, n.iter = 10, n.proc = 12,
                              g1.params = NULL, g2.params = NULL,
                              results = "all",
                              start.params = NULL,
                              dd.model = 2,
                              id.parsopt = c(1,3,4,5),
                              pars.fix = 0,
                              id.parsfix = 2,
                              no.types = 1,
                              nsamples = ntaxa)
```

PB with Speciation/Immigration Linear to DD

Pure-Birth model with equal speciation rates and immigration rate linear to Diversity Dependence This model estimates a single speciation rate shared by the two groups, extinction is set to zero, and the speciation and immigration rates are linearly dependent on the carrying capacity K (linear diversity dependence, ddmodel = 11).

```
eqpb_linDD_11 <- search.surface.DAISIE(model = curr.group, n.iter = 5, n.proc = 12,
                             g1.params = NULL, g2.params = NULL,
                             results = "all",
                             start.params = NULL,
                             dd.model = 11,
                             id.parsopt = c(1,3,4,5),
                             pars.fix = 0,
                             id.parsfix = 2,
                             no.types = 1,
                             nsamples = ntaxa)
```

PB with Speciation/Immigration Exponential to DD

Pure-Birth model with equal speciation rates and immigration rate exponential to Diversity Dependence This model estimates a single speciation rate shared by the two groups, extinction is set to zero, and the speciation and immigration rates are exponentially dependent on the carrying capacity K (exponential diversity dependence, ddmodel = 21).

```
eqpb_expDD_21 <- search.surface.DAISIE(model = curr.group, n.iter = 5, n.proc = 12,
                             g1.params = NULL, g2.params = NULL,
                             results = "all",
                             start.params = NULL,
                             dd.model = 21,
                             id.parsopt = c(1,3,4,5),
                             pars.fix = 0,
                             id.parsfix = 2,
                             no.types = 1,
                             nsamples = ntaxa)
```

Birth Death Models

In all Birth Death models **both** speciation and extinction are estimated from the data.

BD without DD

Birth Death model with equal speciation/extinction rates **without** Diversity Dependence Note: this took ~2 min.

```
eqr_noDD <- search.surface.DAISIE(model = curr.group, n.iter = 10, n.proc = 12,
                             g1.params = NULL, g2.params = NULL,
                             results = "all",
                             start.params = NULL,
                             #start.params = unlist(eqr_noDD_0[c(1,2,4,5)]),
                             dd.model = 0,
                             id.parsopt = c(1,2,4,5),
                             pars.fix = Inf,
                             id.parsfix = 3,
                             no.types = 1,
                             nsamples = ntaxa)
```

BDeq with Speciation Linear to DD

Birth Death model with equal speciation/extinction rates, and specation linear to Diversity Dependence Note: this took ~1 min.

```
eqr_linDD_1 <- search.surface.DAISIE(model = curr.group, n.iter = 10, n.proc = 12,
                             g1.params = NULL, g2.params = NULL,
                             results = "all",
                             start.params = NULL,
                             dd.model = 1,
                             id.parsopt = c(1:5),
                             pars.fix = NULL,
                             id.parsfix = NULL,
                             no.types = 1,
                             nsamples = ntaxa)
```

BDeq with Speciation Exponential to DD

Birth Death model with equal speciation/extinction rates, and speciation exponential to Diversity Dependence Note: this took ~2 min

```
eqr_expDD_2 <- search.surface.DAISIE(model = curr.group, n.iter = 5, n.proc = 12,
                             g1.params = NULL, g2.params = NULL,
                             results = "all",
                             start.params = NULL,
                             dd.model = 2,
                             id.parsopt = c(1:5),
                             pars.fix = NULL,
                             id.parsfix = NULL,
                             no.types = 1,
                             nsamples = ntaxa)
```

BDeq with Speciation/Immigration Linear to DD

Birth Death model with equal speciation/extinction rates, and speciation/immigration linear to Diversity Dependence Note: this took ~35 seconds

```
eqr_linDD_11 <- search.surface.DAISIE(model = curr.group, 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,
                             no.types = 1,
                             nsamples = ntaxa)
```

BDeq with Speciation/Immigration Exponential to DD

Birth Death model with equal speciation/extinction rates, and speciation/immigration exponential to Diversity Dependence Note: this took ~2 min

```
eqr_expDD_21 <- search.surface.DAISIE(model = curr.group, n.iter = 10, n.proc = 12,
                             g1.params = NULL, g2.params = NULL,
                             results = "all",
                             start.params = NULL,
                             dd.model = 21,
                             id.parsopt = c(1:5),
                             pars.fix = NULL,
                             id.parsfix = NULL,
                             no.types = 1,
                             nsamples = ntaxa)
```

Save all the results of the Equal Rates models to a file

Split-Rate Birth Death Models

These models follow the birth death models above, but allow different rates (speciation, extinction, or both) between incumbent and immigrant lineages. They require a DAISIE datalist with two groups designed using *DAISIE_dataprep( )* with *number_clade_types* = 2. We'll have 3 sets of these, allowing (1) **speciation** rates *only* to vary between groups, (2) **extinction** rates *only* to vary between groups, and (3) **speciation** and **extinction** rates to *both* vary between groups. We'll include two additional model sets where speciation rate in the immigrant clade is constrained to 0. One thing to remember is that it's necessary to fix the parameter 11 *p_f* in split-rate models. This is the fraction of mainland species which belong to the second subset rate category. For example, in New Guinea, there are 23 species of Varanus, but 19 of them are in the Hapturosaurus/Euprepiosaurus clade. So when analyzing the Australian dataset, the *p_f* of the immigrant taxa (all Hap/Eup) will be 0.82.

Immigrant Speciation 0 (ImmSp0)

We can start with a very simplistic split rate model where the speciation rate in the immigrant groups is 0.

```
onesp_noDD <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                    g1.params = NULL, g2.params = NULL,
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 0,
                                    id.parsopt = c(1,2,4,5),
                                    pars.fix = c(Inf,0,samp.prop),
                                    id.parsfix = c(3,6,11),
                                    id.parsnoshift = c(7:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

Incumbent Speciation Linear to DD, ImmSp0

```
onesp_linDD_1 <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                    g1.params = NULL, g2.params = NULL,
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 1,
                                    id.parsopt = c(1,2,3,4,5),
                                    pars.fix = c(0,samp.prop),
                                    id.parsfix = c(6,11),
                                    id.parsnoshift = c(7:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

Incumbent Speciation Exponential to DD, ImmSp0

```
onesp_expDD_2 <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                    g1.params = NULL, g2.params = NULL,
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 2,
                                    id.parsopt = c(1,2,3,4,5),
                                    pars.fix = c(0,samp.prop),
                                    id.parsfix = c(6,11),
                                    id.parsnoshift = c(7:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

Incumbent Speciation/Immigration Linear to DD, ImmSp0

```
onesp_linDD_11 <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                        g1.params = NULL, g2.params = NULL,
                                        results = "all",
                                        start.params = NULL,
                                        dd.model = 11,
                                        id.parsopt = c(1,2,3,4,5),
                                        pars.fix = c(0,samp.prop),
                                        id.parsfix = c(6,11),
                                        id.parsnoshift = c(7:10),
                                        no.types = 2,
                                        nsamples = ntaxa)
```

Incumbent Speciation/Immigration Exponential to DD, ImmSp0

```
onesp_expDD_21 <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                        g1.params = NULL, g2.params = NULL,
                                        results = "all",
                                        start.params = NULL,
                                        dd.model = 21,
                                        id.parsopt = c(1,2,3,4,5),
                                        pars.fix = c(0,samp.prop),
                                        id.parsfix = c(6,11),
                                        id.parsnoshift = c(7:10),
                                        no.types = 2,
                                        nsamples = ntaxa)
```

BD2sp without DD

Birth Death model with **two** speciation rates **without** 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). Note: this took ~2 min

```
twosp_noDD <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 0,
                                    id.parsopt = c(1,2,4,5,6),
                                    pars.fix = c(Inf,samp.prop),
                                    id.parsfix = c(3,11),
                                    id.parsnoshift = c(7:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2sp Speciation Linear to DD

BD model with **two** speciation rates linear to Diversity Dependence Note: this took ~45 sec

```
twosp_linDD_1 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 1,
                                    id.parsopt = c(1:6),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(7:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2sp Speciation Exponential to DD

BD model with **two** speciation rates exponential to Diversity Dependence Note: this took ~2 min

```
twosp_expDD_2 <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 2,
                                    id.parsopt = c(1:6),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(7:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2sp Speciation/Immigration Linear to DD

BD model with **two** speciation/immigration rates linear to Diversity Dependence Note: this took ~1 min

```
twosp_linDD_11 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 11,
                                    id.parsopt = c(1:6),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(7:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2sp Speciation/Immigration Exponential to DD

BD model with **two** speciation/immigration rates exponential to Diversity Dependence Note: this took ~2.5 min

```
twosp_expDD_21 <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 21,
                                    id.parsopt = c(1:6),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(7:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

Save all the results of the Two Sp Rates models to a file

BD2ex without DD

BD model with two extinction rates (equal speciation rates) and no Diversity Dependence ~35 min

```
twoex_noDD <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    # becuase the way I've got the 'init.params' bit coded
                                    # to optimize a separate extinction but not speciation
                                    # you still have to designate speciation in the g2.params
                                    # but speciation is listed in the 'id.parsnoshift' set
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 11,
                                    id.parsopt = c(1,2,4,5,7),
                                    pars.fix = c(Inf, samp.prop),
                                    id.parsfix = c(3,11),
                                    id.parsnoshift = c(6,8,9,10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2ex with Speciation Linear to DD

BD model with two extinction rates (equal speciation rates) and speciation rate linear to Diversity Depend

1 min

```
twoex_linDD_1 <- search.surface.DAISIE(model = curr.split, n.iter = 10, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    # becuase the way I've got the 'init.params' bit coded
                                    # to optimize a separate extinction but not speciation
                                    # you still have to designate speciation in the g2.params
                                    # but speciation is listed in the 'id.parsnoshift' set
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 1,
                                    id.parsopt = c(1,2,3,4,5,7),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(6,8,9,10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2ex with Speciation Exponential to DD

BD model with two extinction rates (equal speciation rates) and speciation rate exponential to Diversity Dependence ~1.5 min

```
twoex_expDD_2 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    # becuase the way I've got the 'init.params' bit coded
                                    # to optimize a separate extinction but not speciation
                                    # you still have to designate speciation in the g2.params
                                    # but speciation is listed in the 'id.parsnoshift' set
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 2,
                                    id.parsopt = c(1,2,3,4,5,7),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(6,8,9,10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2ex with Speciation/Immigration Linear to DD

BD model with two extinction rates (equal speciation rates) and speciation/immigration rates linear to Div

1min

```
twoex_linDD_11 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    # becuase the way I've got the 'init.params' bit coded
                                    # to optimize a separate extinction but not speciation
                                    # you still have to designate speciation in the g2.params
                                    # but speciation is listed in the 'id.parsnoshift' set
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 11,
                                    id.parsopt = c(1,2,3,4,5,7),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(6,8,9,10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2ex with Speciation/Immigration Exponential to DD

BD model with two extinction rates (equal speciation rates) and speciation/immigration rates linear to Div

2 min

```
twoex_expDD_21 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    # becuase the way I've got the 'init.params' bit coded
                                    # to optimize a separate extinction but not speciation
                                    # you still have to designate speciation in the g2.params
                                    # but speciation is listed in the 'id.parsnoshift' set
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 21,
                                    id.parsopt = c(1,2,3,4,5,7),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(6,8,9,10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

Save all the results of the Two Ex Rates models to a file

Split-Rate Birth Death Models with varying speciation and extinction

ImmSp0, BD2ex, noDD

Two extinction rates, and speciation rate in immigrants is fixed to 0 with no diversity dependence. 45 sec

```
twoex_immsp0_noDD <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 0,
                                    id.parsopt = c(1,2,4,5,7),
                                    pars.fix = c(Inf,0,samp.prop),
                                    id.parsfix = c(3,6,11),
                                    id.parsnoshift = c(8:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

ImmSp0, BD2ex, Speciation Linear to DD

~ 40 sec

```
twoex_immsp0_linDD_1 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 1,
                                    id.parsopt = c(1,2,3,4,5,7),
                                    pars.fix = c(0,samp.prop),
                                    id.parsfix = c(6,11),
                                    id.parsnoshift = c(8:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

ImmSp0, BD2ex, Speciation Exponential to DD

~ 1.5 sec

```
twoex_immsp0_expDD_2 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 2,
                                    id.parsopt = c(1,2,3,4,5,7),
                                    pars.fix = c(0,samp.prop),
                                    id.parsfix = c(6,11),
                                    id.parsnoshift = c(8:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

ImmSp0, BD2ex, Speciation/Immigration Linear to DD

~ 45 sec

```
twoex_immsp0_linDD_11 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 11,
                                    id.parsopt = c(1,2,3,4,5,7),
                                    pars.fix = c(0,samp.prop),
                                    id.parsfix = c(6,11),
                                    id.parsnoshift = c(8:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

ImmSp0, BD2ex, Speciation/Immigration Exponential to DD

~ 1.7 min

```
twoex_immsp0_expDD_21 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                               g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                               results = "all",
                                               start.params = NULL,
                                               dd.model = 21,
                                               id.parsopt = c(1,2,3,4,5,7),
                                               pars.fix = c(0,samp.prop),
                                               id.parsfix = c(6,11),
                                               id.parsnoshift = c(8:10),
                                               no.types = 2,
                                               nsamples = ntaxa)
```

BD2spex without DD

BD model with two speciation and extinction rates and **without** Diversity Dependence Note: this took ~ 1 min

```
twospex_noDD <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 0,
                                    id.parsopt = c(1,2,4,5,6,7),
                                    pars.fix = c(Inf,samp.prop),
                                    id.parsfix = c(3,11),
                                    id.parsnoshift = c(8:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2spex with Speciation Linear to DD

BD model with two speciation and extinction rates and speciation rates linear to Diversity Dependence ~1.2 min

```
twospex_linDD_1 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 1,
                                    id.parsopt = c(1:7),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(8:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2spex with Speciation Exponential to DD

BD model with two speciation and extinction rates and speciation rates exponential to Diversity Dependen

2.4 min

```
twospex_expDD_2 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                    g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                    results = "all",
                                    start.params = NULL,
                                    dd.model = 2,
                                    id.parsopt = c(1:7),
                                    pars.fix = samp.prop,
                                    id.parsfix = 11,
                                    id.parsnoshift = c(8:10),
                                    no.types = 2,
                                    nsamples = ntaxa)
```

BD2spex with Speciation/Immigration Linear to DD

BD model with two speciation and extinction rates and speciation/immigration rates linear to Diversity De

42 sec

```
twospex_linDD_11 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                          g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                          results = "all",
                                          start.params = NULL,
                                          dd.model = 11,
                                          id.parsopt = c(1:7),
                                          pars.fix = samp.prop,
                                          id.parsfix = 11,
                                          id.parsnoshift = c(8:10),
                                          no.types = 2,
                                          nsamples = ntaxa)
```

BD2spex with Speciation/Immigration Exponential to DD

BD model with two speciation and extinction rates and speciation/immigration rates exponential to Divers

2 min

```
twospex_expDD_21 <- search.surface.DAISIE(model = curr.split, n.iter = 5, n.proc = 12,
                                          g1.params = NULL, g2.params = c("lambda_c", "mu"),
                                          results = "all",
                                          start.params = NULL,
                                          dd.model = 21,
                                          id.parsopt = c(1:7),
                                          pars.fix = samp.prop,
                                          id.parsfix = 11,
                                          id.parsnoshift = c(8:10),
                                          no.types = 2,
                                          nsamples = ntaxa)
```

Save all the results of the Two Sp/Ex Rates models to a file

Save all the results of the Two Ex Rates models to a file

Make sure you've got all the models saved somewhere safe

```
#all.models <- list(pb_res, eqr_res, twosp_res, twospex_res, twoex_res)
#saveRDS(all.models, "../Results/DAISIE_Australian_AllModel_Results.RDS")

# MAKE SURE THAT YOU CHANGE THE NAME OF THE OUTPUT FILE!
save(eqpb_noDD, eqpb_linDD_1, eqpb_linDD_11, eqpb_expDD_2, eqpb_expDD_21,
     eqr_noDD, eqr_linDD_1, eqr_linDD_11,eqr_expDD_2, eqr_expDD_21,
     onesp_noDD, onesp_linDD_1, onesp_linDD_11, onesp_expDD_2, onesp_expDD_21,
     twosp_noDD, twosp_linDD_1, twosp_linDD_11, twosp_expDD_2, twosp_expDD_21,
     twoex_noDD, twoex_linDD_1, twoex_linDD_11, twoex_expDD_2, twoex_expDD_21,
     twoex_immsp0_noDD, twoex_immsp0_linDD_1, twoex_immsp0_linDD_11,
     twoex_immsp0_expDD_2, twoex_immsp0_expDD_21,
     twospex_noDD, twospex_linDD_1, twospex_linDD_11,
     twospex_expDD_2, twospex_expDD_21,
     file="../Results/DAISIE_NewGuinea_AllModel_Results.RData")
```

Comparing Empirical Model Fits

Start by designating which models we want to compare.

```
all.DAISIE.models <- c("eqpb_noDD", "eqpb_linDD_1", "eqpb_linDD_11",
                       "eqpb_expDD_2", "eqpb_expDD_21",
                       "eqr_noDD", "eqr_linDD_1", "eqr_linDD_11",
                       "eqr_expDD_2", "eqr_expDD_21",
                       "onesp_noDD", "onesp_linDD_1", "onesp_linDD_11",
                       "onesp_expDD_2", "onesp_expDD_21",
                       "twosp_noDD", "twosp_linDD_1", "twosp_linDD_11",
                       "twosp_expDD_2", "twosp_expDD_21",
                       "twospex_noDD", "twospex_linDD_1", "twospex_linDD_11",
                       "twospex_expDD_2", "twospex_expDD_21",
                       "twoex_noDD", "twoex_linDD_1", "twoex_linDD_11",
                       "twoex_expDD_2", "twoex_expDD_21",
                       "twoex_immsp0_noDD", "twoex_immsp0_linDD_1",
                       "twoex_immsp0_linDD_11", "twoex_immsp0_expDD_2",
                       "twoex_immsp0_expDD_21")
```

Model Fits with AIC

Use the *DAISIE.AIC* function to compare the model fits with AIC
This will take the model names as input, use the likelihood to calculate AIC and corrected AIC (AICc), and tell us the contribution of each model to the total model weights (AICcwt).

```
all.AIC <- DAISIE.AIC(models = all.DAISIE.models,
                      method="best")
```

Model Fits with BIC

Use the *DAISIE.BIC* function to compare the model fits with BIC
This will take the model as input, use the likelihood to calculate BIC, and tell us the contribution of each model to the total model weights (BICwt)

```
all.BIC <- DAISIE.BIC(models = all.DAISIE.models,
                      method="best")
```

Australia Results

Let's check model comparison using AIC first

```
all.AIC <- DAISIE.AIC(models = all.DAISIE.models,
                      method="best")
```

```
##                           logLik no.Param     AICc delta.AICc       AICcwt
## onesp_linDD_11         -111.3750        5 234.8929   0.000000 1.542773e-01
## onesp_linDD_1          -111.4607        5 235.0642   0.171353 1.416098e-01
## onesp_expDD_21         -111.9070        5 235.9569   1.064055 9.062447e-02
## twoex_immsp0_linDD_11 -110.4407        6 235.9925   1.099611 8.902755e-02
## onesp_expDD_2          -112.0028        5 236.1485   1.255660 8.234531e-02
## twoex_immsp0_linDD_1  -110.5714        6 236.2540   1.361101 7.811648e-02
## twoex_immsp0_expDD_21 -110.7885        6 236.6882   1.795322 6.287134e-02
## twoex_immsp0_expDD_2  -110.9427        6 236.9964   2.103587 5.389072e-02
## twosp_linDD_11         -111.3750        6 237.8611   2.968273 3.497435e-02
## twosp_linDD_1          -111.4607        6 238.0325   3.139607 3.210296e-02
## twoex_linDD_11         -111.7824        6 238.6758   3.782958 2.327250e-02
## twosp_expDD_21         -111.9070        6 238.9252   4.032309 2.054458e-02
## twoex_linDD_1          -111.9283        6 238.9678   4.074941 2.011129e-02
## twosp_expDD_2          -112.0028        6 239.1168   4.223915 1.866769e-02
## twospex_linDD_11       -110.4407        7 239.1891   4.296203 1.800501e-02
## twospex_linDD_1        -110.5714        7 239.4505   4.557684 1.579842e-02
## twoex_expDD_21         -112.1769        6 239.4650   4.572094 1.568500e-02
## twoex_expDD_2          -112.3475        6 239.8060   4.913175 1.322573e-02
## twospex_expDD_21       -110.7885        7 239.8848   4.991913 1.271516e-02
## twospex_expDD_2        -110.9427        7 240.1930   5.300171 1.089895e-02
## twoex_immsp0_noDD      -115.5668        5 243.2764   8.383571 2.332560e-03
## twoex_noDD             -115.9666        5 244.0760   9.183162 1.563881e-03
## eqpb_expDD_21          -117.7216        4 244.8224   9.929563 1.076774e-03
## eqpb_expDD_2           -117.7558        4 244.8910   9.998125 1.040487e-03
## eqpb_linDD_11          -117.9729        4 245.3252  10.432321 8.374371e-04
## eqpb_linDD_1           -117.9986        4 245.3765  10.483588 8.162437e-04
## eqr_linDD_11           -116.8097        5 245.7623  10.869450 6.730245e-04
## eqr_linDD_1            -116.8427        5 245.8283  10.935429 6.511842e-04
## twospex_noDD           -115.5668        6 246.2447  11.351847 5.287858e-04
## eqr_expDD_21           -117.2666        5 246.6761  11.783240 4.261908e-04
## eqr_expDD_2            -117.3040        5 246.7509  11.857991 4.105557e-04
## onesp_noDD             -118.8030        4 246.9853  12.092430 3.651439e-04
## eqpb_noDD              -120.1924        3 247.1848  12.291918 3.304805e-04
## twosp_noDD             -118.8030        5 249.7488  14.855978 9.169958e-05
## eqr_noDD               -120.1924        4 249.7641  14.871232 9.100286e-05
```

```
##                  logLik no.Param    AIC     AICc delta.AICc    AICcwt
## onesp_linDD_11 -111.375        5 232.75 234.8929          0 0.1542773
```

```
##    lambda_c        mu       K      gamma  lambda_a lambda_c2       mu2      K2
## 1 0.6639816 0.1215439 36.3146 0.01209569 0.3180357         0 0.1215439 36.3146
##       gamma2 lambda_a2 prop_type2   loglik df conv
## 1 0.01209569 0.3180357       0.82 -111.375  5    0
```

Then with BIC next

```
all.BIC <- DAISIE.BIC(models = all.DAISIE.models,
                      method="best")
```

```
## $results
##                          logLik no.Param      BIC delta.BIC        BICwt
## onesp_linDD_11        -111.3750        5 240.3818  0.000000 0.1775179383
## onesp_linDD_1         -111.4607        5 240.5532  0.171353 0.1629421421
## onesp_expDD_21        -111.9070        5 241.4459  1.064055 0.1042763295
## onesp_expDD_2         -112.0028        5 241.6375  1.255660 0.0947499832
## twoex_immsp0_linDD_11 -110.4407        6 242.0395  1.657718 0.0774949399
## twoex_immsp0_linDD_1  -110.5714        6 242.3010  1.919207 0.0679972949
## twoex_immsp0_expDD_21 -110.7885        6 242.7352  2.353429 0.0547270037
## twoex_immsp0_expDD_2  -110.9427        6 243.0435  2.661693 0.0469097309
## twosp_linDD_11        -111.3750        6 243.9082  3.526380 0.0304437854
## twosp_linDD_1         -111.4607        6 244.0795  3.697713 0.0279443470
## twoex_linDD_11        -111.7824        6 244.7229  4.341065 0.0202577859
## twosp_expDD_21        -111.9070        6 244.9722  4.590415 0.0178832417
## twoex_linDD_1         -111.9283        6 245.0149  4.633047 0.0175060747
## twosp_expDD_2         -112.0028        6 245.1638  4.782021 0.0162494810
## twoex_expDD_21        -112.1769        6 245.5120  5.130200 0.0136531662
## twospex_linDD_11      -110.4407        7 245.5659  5.184088 0.0132902070
## twospex_linDD_1       -110.5714        7 245.8274  5.445570 0.0116614311
## twoex_expDD_2         -112.3475        6 245.8531  5.471282 0.0115124700
## twospex_expDD_21      -110.7885        7 246.2616  5.879799 0.0093855578
## twospex_expDD_2       -110.9427        7 246.5699  6.188057 0.0080449393
## twoex_immsp0_noDD     -115.5668        5 248.7654  8.383571 0.0026839413
## eqpb_expDD_21         -117.7216        4 249.5486  9.166749 0.0018142949
## twoex_noDD            -115.9666        5 249.5650  9.183162 0.0017994677
## eqpb_expDD_2          -117.7558        4 249.6171  9.235311 0.0017531536
## eqpb_linDD_11         -117.9729        4 250.0513  9.669508 0.0014110271
## eqpb_linDD_1          -117.9986        4 250.1026  9.720774 0.0013753176
## eqpb_noDD             -120.1924        3 250.9639 10.582054 0.0008940841
## eqr_linDD_11          -116.8097        5 251.2513 10.869450 0.0007744104
## eqr_linDD_1           -116.8427        5 251.3172 10.935429 0.0007492799
## onesp_noDD            -118.8030        4 251.7114 11.329617 0.0006152438
## eqr_expDD_21          -117.2666        5 252.1650 11.783240 0.0004903930
## eqr_expDD_2           -117.3040        5 252.2398 11.857991 0.0004724026
## twospex_noDD          -115.5668        6 252.2918 11.909953 0.0004602871
## eqr_noDD              -120.1924        4 254.4902 14.108418 0.0001533339
## twosp_noDD            -118.8030        5 255.2378 14.855978 0.0001055134
##
## $best.model
##                  logLik no.Param      BIC delta.BIC     BICwt
## onesp_linDD_11 -111.375        5 240.3818         0 0.1775179
##
## $parameter.estimates
##    lambda_c        mu       K      gamma  lambda_a lambda_c2       mu2      K2
## 1 0.6639816 0.1215439 36.3146 0.01209569 0.3180357         0 0.1215439 36.3146
##       gamma2 lambda_a2 prop_type2   loglik df conv
## 1 0.01209569 0.3180357       0.82 -111.375  5    0
```

Results for the New Guinea Monitors

AIC first

```
all.AIC <- DAISIE.AIC(models = all.DAISIE.models,
                      method="best")
```

```
##                          logLik no.Param     AICc delta.AICc       AICcwt
## onesp_noDD            -68.04250        4 146.3072   0.000000 3.533914e-01
## onesp_linDD_11        -67.70894        5 148.9473   2.640059 9.440052e-02
## onesp_linDD_1         -67.72944        5 148.9883   2.681063 9.248483e-02
## onesp_expDD_21        -67.78043        5 149.0903   2.783051 8.788692e-02
## onesp_expDD_2         -67.79979        5 149.1290   2.821771 8.620179e-02
## twoex_immsp0_noDD     -68.02561        5 149.5806   3.273407 6.877724e-02
## twosp_noDD            -68.04251        5 149.6144   3.307196 6.762505e-02
## twoex_immsp0_linDD_11 -67.61606        6 152.4821   6.174896 1.612109e-02
## twoex_immsp0_linDD_1  -67.64048        6 152.5310   6.223738 1.573217e-02
## twosp_linDD_11        -67.70894        6 152.6679   6.360656 1.469120e-02
## twoex_immsp0_expDD_21 -67.72730        6 152.7046   6.397371 1.442396e-02
## twosp_linDD_1         -67.72945        6 152.7089   6.401672 1.439298e-02
## twoex_immsp0_expDD_2  -67.75159        6 152.7532   6.445954 1.407780e-02
## twosp_expDD_21        -67.78043        6 152.8109   6.503639 1.367756e-02
## twosp_expDD_2         -67.79979        6 152.8496   6.542360 1.341531e-02
## twospex_noDD          -68.02561        6 153.3012   6.993996 1.070358e-02
## twoex_noDD            -70.25232        5 154.0341   7.726825 7.419887e-03
## twospex_linDD_11      -67.61607        7 156.6988  10.391574 1.957738e-03
## twospex_linDD_1       -67.64048        7 156.7476  10.440407 1.910515e-03
## twospex_expDD_21      -67.72730        7 156.9213  10.614040 1.751647e-03
## twospex_expDD_2       -67.75159        7 156.9698  10.662616 1.709615e-03
## eqr_noDD              -73.69691        4 157.6160  11.308814 1.237594e-03
## twoex_linDD_1         -70.25232        6 157.7546  11.447412 1.154735e-03
## twoex_expDD_21        -70.25232        6 157.7546  11.447419 1.154730e-03
## twoex_linDD_11        -70.25233        6 157.7547  11.447424 1.154728e-03
## twoex_expDD_2         -70.25233        6 157.7547  11.447432 1.154723e-03
## eqr_expDD_21          -73.69691        5 160.9232  14.616004 2.368270e-04
## eqr_linDD_1           -73.69693        5 160.9233  14.616052 2.368213e-04
## eqr_linDD_11          -73.69694        5 160.9233  14.616069 2.368193e-04
## eqr_expDD_2           -73.69697        5 160.9234  14.616130 2.368120e-04
## eqpb_noDD             -77.02232        3 161.3078  15.000570 1.953996e-04
## eqpb_expDD_21         -77.02232        4 163.4240  17.116723 6.782767e-05
## eqpb_linDD_11         -77.02232        4 163.4240  17.116723 6.782767e-05
## eqpb_expDD_2          -77.02232        4 163.4240  17.116725 6.782758e-05
## eqpb_linDD_1          -77.02232        4 164.2669  17.959635 4.450111e-05
```

```
##              logLik no.Param     AIC     AICc delta.AICc    AICcwt
## onesp_noDD -68.0425        4 144.085 146.3072          0 0.3533914
```

```
##    lambda_c        mu   K      gamma   lambda_a lambda_c2       mu2  K2
## 1 0.3619297 0.1373165 Inf 0.01144121 0.07913357         0 0.1373165 Inf
##       gamma2  lambda_a2 prop_type2   loglik df conv
## 1 0.01144121 0.07913357        0.9 -68.0425  4    0
```

BIC next

```
all.BIC <- DAISIE.BIC(models = all.DAISIE.models,
                      method="best")
```

```
## $results
##                          logLik no.Param      BIC delta.BIC        BICwt
## onesp_noDD            -68.04250        4 148.6270  0.000000 3.164040e-01
## onesp_linDD_11        -67.70894        5 151.0953  2.468364 9.209658e-02
## onesp_linDD_1         -67.72944        5 151.1364  2.509368 9.022764e-02
## onesp_expDD_21        -67.78043        5 151.2383  2.611355 8.574195e-02
## onesp_expDD_2         -67.79979        5 151.2771  2.650075 8.409795e-02
## twoex_immsp0_noDD     -68.02561        5 151.7287  3.101711 6.709867e-02
## twosp_noDD            -68.04251        5 151.7625  3.135500 6.597459e-02
## twoex_immsp0_linDD_11 -67.61606        6 154.0451  5.418106 2.107246e-02
## twoex_immsp0_linDD_1  -67.64048        6 154.0939  5.466948 2.056408e-02
## twosp_linDD_11        -67.70894        6 154.2308  5.603866 1.920339e-02
## twoex_immsp0_expDD_21 -67.72730        6 154.2676  5.640582 1.885407e-02
## twosp_linDD_1         -67.72945        6 154.2719  5.644882 1.881357e-02
## twoex_immsp0_expDD_2  -67.75159        6 154.3161  5.689165 1.840160e-02
## twosp_expDD_21        -67.78043        6 154.3738  5.746850 1.787843e-02
## twosp_expDD_2         -67.79979        6 154.4126  5.785570 1.753563e-02
## twospex_noDD          -68.02561        6 154.8642  6.237206 1.399104e-02
## twoex_noDD            -70.25232        5 156.1821  7.555130 7.238797e-03
## twospex_linDD_11      -67.61607        7 157.1806  8.553612 4.393886e-03
## twospex_linDD_1       -67.64048        7 157.2294  8.602446 4.287900e-03
## twospex_expDD_21      -67.72730        7 157.4031  8.776078 3.931343e-03
## twospex_expDD_2       -67.75159        7 157.4516  8.824655 3.837007e-03
## twoex_linDD_1         -70.25232        6 159.3176 10.690622 1.509395e-03
## twoex_expDD_21        -70.25232        6 159.3176 10.690630 1.509389e-03
## twoex_linDD_11        -70.25233        6 159.3176 10.690634 1.509386e-03
## twoex_expDD_2         -70.25233        6 159.3176 10.690643 1.509379e-03
## eqr_noDD              -73.69691        4 159.9358 11.308814 1.108063e-03
## eqr_expDD_21          -73.69691        5 163.0713 14.444308 2.310470e-04
## eqr_linDD_1           -73.69693        5 163.0713 14.444356 2.310414e-04
## eqr_linDD_11          -73.69694        5 163.0714 14.444373 2.310395e-04
## eqr_expDD_2           -73.69697        5 163.0714 14.444434 2.310324e-04
## eqpb_noDD             -77.02232        3 163.4511 14.824140 1.910825e-04
## eqpb_linDD_1          -77.02232        4 166.5866 17.959635 3.984343e-05
## eqpb_expDD_21         -77.02232        4 168.1501 19.523100 1.823286e-05
## eqpb_linDD_11         -77.02232        4 168.1501 19.523100 1.823286e-05
## eqpb_expDD_2          -77.02232        4 168.1501 19.523102 1.823283e-05
## 
## $best.model
##              logLik no.Param     BIC delta.BIC    BICwt
## onesp_noDD -68.0425        4 148.627         0 0.316404
## 
## $parameter.estimates
##    lambda_c        mu   K      gamma   lambda_a lambda_c2       mu2  K2
## 1 0.3619297 0.1373165 Inf 0.01144121 0.07913357         0 0.1373165 Inf
##       gamma2  lambda_a2 prop_type2   loglik df conv
## 1 0.01144121 0.07913357        0.9 -68.0425  4    0
```

DAISIE Models Explained

Model Categories

Q	Regimes	Model Type	Model Name	Shared	Varied
1	1	Pure Birth	Pure Birth—*eqpb*	λ^{c}, γ, λ^{a}, K	—
2	1	Birth Death	Equal Rates—*eqr*	λ^{c}, μ, γ, λ^{a}, K	—
3	2	Split Rate Birth Death	Split λ—*onesp*	μ, γ, λ^{a}, K	λ^{c} (λ^{c}_{2}=0)
4	2	Split Rate Birth Death	Split λ—*twosp*	μ, γ, λ^{a}, K	λ^{c}
5	2	Split Rate Birth Death	Split μ—*twoex*	λ^{c}, γ, λ^{a}, K	μ
6	2	Split Rate Birth Death	Split μ—*twoex__immsp0*	γ, λ^{a}, K	λ^{c} (λ^{c}_{2}=0), μ
7	2	Split Rate Birth Death	Split Sp/Ex—*twospex*	γ, λ^{a}, K	λ^{c}, μ

Model Specifications

Model#	Q	DAISIE Model	Model Name	DD	#Par.	λ-DD Corr.	γ-DD Corr.	Fixed Parameters	Free Parameters
M1	1	*eqpb__noDD*	Pure Birth	0	3	—	—	K=Inf	λ^{c}, γ, λ^{a}
M2	1	*eqpb__linDD__1*	Pure Birth	1	4	linear	—		λ^{c}, γ, λ^{a}, K
M3	1	*eqpb__linDD__11*	Pure Birth	11	4	linear	linear		λ^{c}, γ, λ^{a}, K
M4	1	*eqpb__expDD__2*	Pure Birth	2	4	exponential	—		λ^{c}, γ, λ^{a}, K
M5	1	*eqpb__expDD__21*	Pure Birth	21	4	exponential	exponential		λ^{c}, γ, λ^{a}, K
M6	2	*eqr__noDD*	Equal Rates	0	4	—	—	K=Inf	λ^{c}, μ, γ, λ^{a}
M7	2	*eqr__linDD__1*	Equal Rates	1	5	linear	—		λ^{c}, μ, γ, λ^{a}, K
M8	2	*eqr__linDD__11*	Equal Rates	11	5	linear	linear		λ^{c}, μ, γ, λ^{a}, K
M9	2	*eqr__expDD__2*	Equal Rates	2	5	exponential	—		λ^{c}, μ, γ, λ^{a}, K
M10	2	*eqr__expDD__21*	Equal Rates	21	5	exponential	exponential		λ^{c}, μ, γ, λ^{a}, K
M11	3	*onesp__noDD*	Split λ	0	4	—	—	K=Inf, λ_{c2}=0	λ^{c}_{1}, μ, γ, λ^{a}
M12	3	*onesp__linDD__1*	Split λ	1	5	linear	—	λ_{c2}=0	λ^{c}_{1}, μ, γ, λ^{a}, K
M13	3	*onesp__linDD__11*	Split λ	11	5	linear	linear	λ_{c2}=0	λ^{c}_{1}, μ, γ, λ^{a}, K
M14	3	*onesp__expDD__2*	Split λ	2	5	exponential	—	λ_{c2}=0	λ^{c}_{1}, μ, γ, λ^{a}, K
M15	3	*onesp__expDD__21*	Split λ	21	5	exponential	exponential	λ_{c2}=0	λ^{c}_{1}, μ, γ, λ^{a}, K
M16	4	*twosp__noDD*	Split λ	0	5	—	—	K=Inf	λ^{c}_{1}, μ, γ, λ^{a}, λ^{c}_{2}
M17	4	*twosp__linDD__1*	Split λ	1	6	linear	—		λ^{c}_{1}, μ, γ, λ^{a}, K, λ^{c}_{2}
M18	4	*twosp__linDD__11*	Split λ	11	6	linear	linear		λ^{c}_{1}, μ, γ, λ^{a}, K, λ^{c}_{2}
M19	4	*twosp__expDD__2*	Split λ	2	6	exponential	—		λ^{c}_{1}, μ, γ, λ^{a}, K, λ^{c}_{2}
M20	4	*twosp__expDD__21*	Split λ	21	6	exponential	exponential		λ^{c}_{1}, μ, γ, λ^{a}, K, λ^{c}_{2}
M21	5	*twoex__noDD*	Split μ	0	5	—	—	K=Inf	λ^{c}, μ_{1}, γ, λ^{a}, μ_{2}

Model#	Q	DAISIE Model	Model Name	DD	#Par.	λ-DD Corr.	γ-DD Corr.	Fixed Parameters	Free Parameters
M22	5	*twoex_linDD_1*	Split μ	1	6	linear	—		λ^{c}, μ_1, γ, λ^{a}, K, μ_2
M23	5	*twoex_linDD_11*	Split μ	11	6	linear	linear		λ^{c}, μ_1, γ, λ^{a}, K, μ_2
M24	5	*twoex_expDD_2*	Split μ	2	6	exponential	—		λ^{c}, μ_1, γ, λ^{a}, K, μ_2
M25	5	*twoex_expDD_21*	Split μ	21	6	exponential	exponential		λ^{c}, μ_1, γ, λ^{a}, K, μ_2
M26	6	*twoex_immsp0_noDD*	Split λ, μ	0	5	—	—	K=Inf, $\lambda_{\mathrm{c}2}$=0	λ^{c}, μ_1, γ, λ^{a}, μ_2
M27	6	*twoex_immsp0_linDD_1*	Split λ, μ	1	6	linear	—	$\lambda_{\mathrm{c}2}$=0	λ^{c}, μ_1, γ, λ^{a}, K, μ_2
M28	6	*twoex_immsp0_linDD_11*	Split λ, μ	11	6	linear	linear	$\lambda_{\mathrm{c}2}$=0	λ^{c}, μ_1, γ, λ^{a}, K, μ_2
M29	6	*twoex_immsp0_expDD_2*	Split λ, μ	2	6	exponential	—	$\lambda_{\mathrm{c}2}$=0	λ^{c}, μ_1, γ, λ^{a}, K, μ_2
M30	6	*twoex_immsp0_linDD_21*	Split λ, μ	1	6	exponential	exponential	$\lambda_{\mathrm{c}2}$=0	λ^{c}, μ_1, γ, λ^{a}, K, μ_2
M31	7	*twospex_noDD*	Split λ, μ	0	6	—	—	K=Inf	$\lambda^{\mathrm{c}}{}_1$, μ_1, γ, λ^{a}, $\lambda^{\mathrm{c}}{}_2$, μ_2
M32	7	*twospex_linDD_1*	Split λ, μ	1	7	linear	—		$\lambda^{\mathrm{c}}{}_1$, μ_1, γ, λ^{a}, K, $\lambda^{\mathrm{c}}{}_2$, μ_2
M33	7	*twospex_linDD_11*	Split λ, μ	11	7	linear	linear		$\lambda_{\mathrm{c}1}$, μ_1, γ, λ^{a}, K, $\lambda^{\mathrm{c}}{}_2$, μ_2
M34	7	*twospex_expDD_2*	Split λ, μ	2	7	exponential	—		$\lambda_{\mathrm{c}1}$, μ_1, γ, λ^{a}, K, $\lambda^{\mathrm{c}}{}_2$, μ_2
M35	7	*twospex_expDD_21*	Split λ, μ	21	7	exponential	exponential		$\lambda_{\mathrm{c}1}$, μ_1, γ, λ^{a}, K, $\lambda^{\mathrm{c}}{}_2$, μ_2

Diversification Models with ClaDS

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

Load RPANDA

```
library(RPANDA)
```

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

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

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)])
```

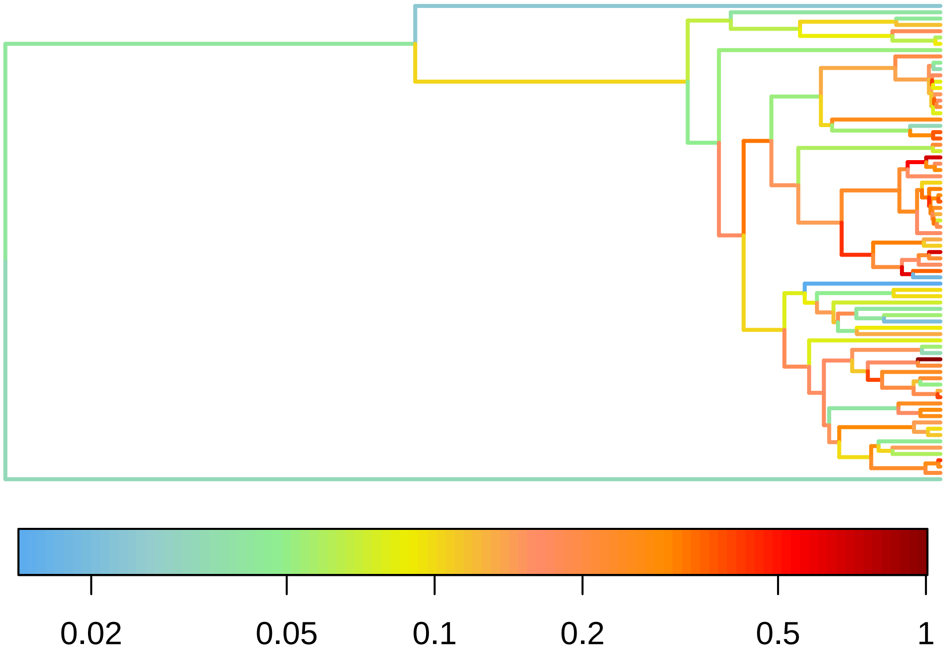
Here's an example file I didn't run long enough:

```
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)])
```

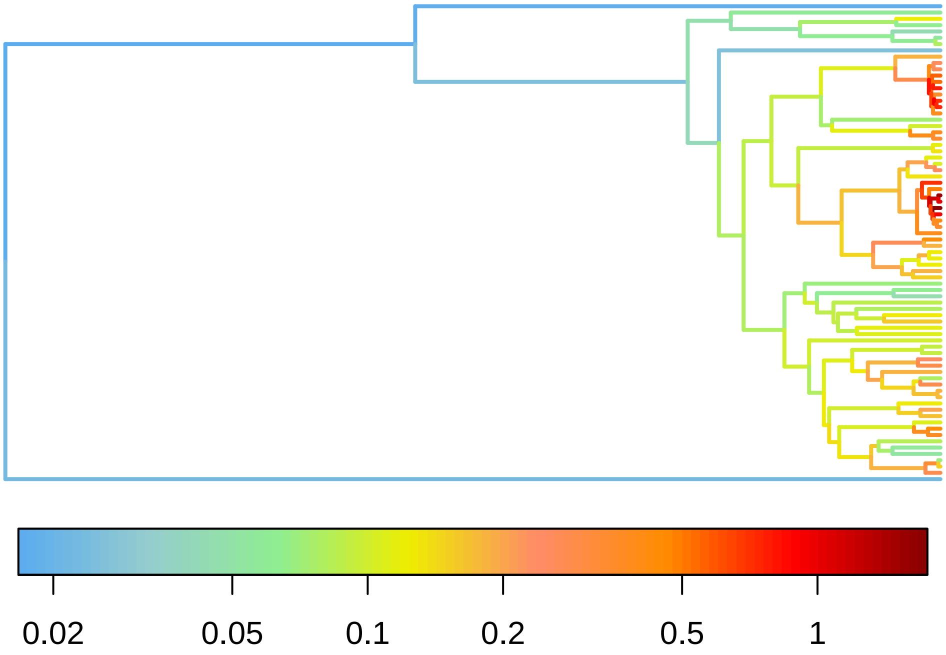


The above example was run for 10,000 generations, thin = 100. Run it longer by just providing the previous file

```
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)
```

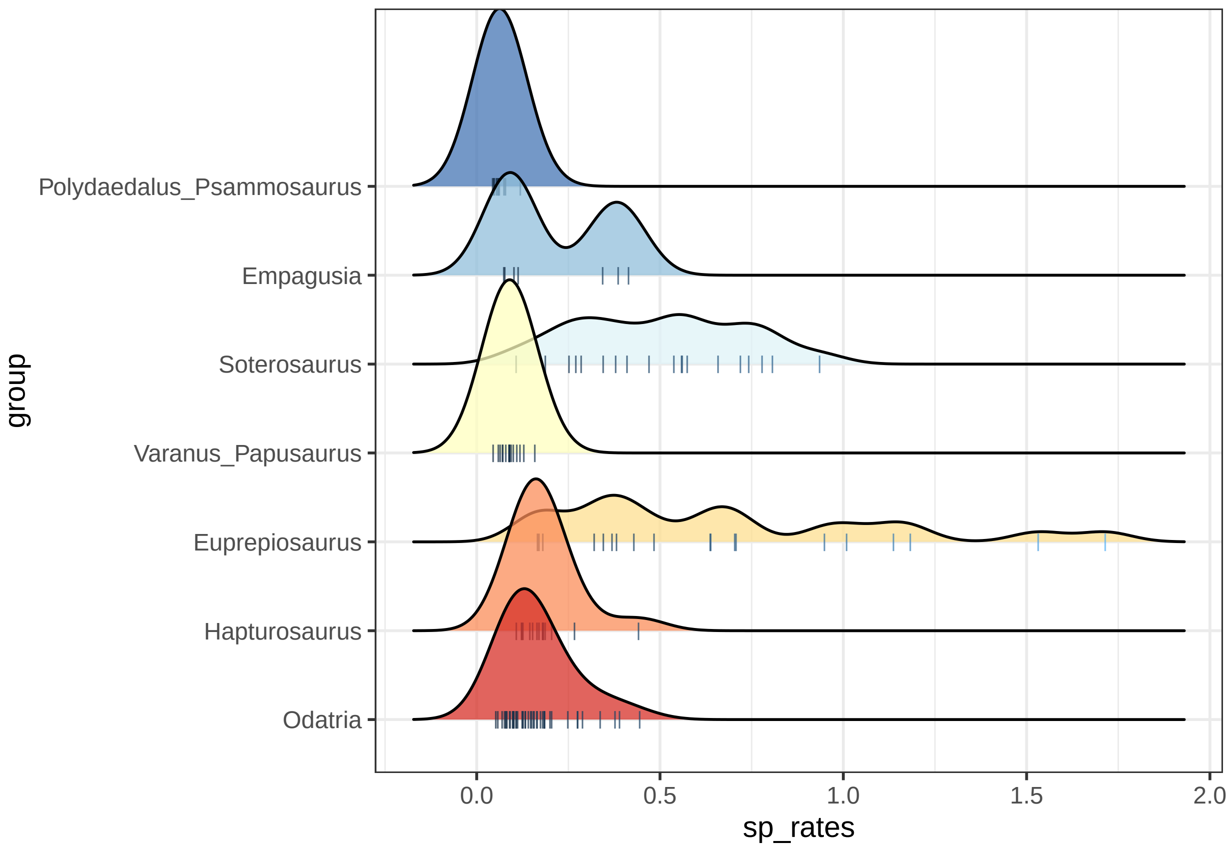
```
euprep.rates  <- extract.ClaDS.rates(maps, 83, 101)
varanus.rates <- extract.ClaDS.rates(maps, 50, 66)
sotero.rates  <- extract.ClaDS.rates(maps, 119, 137)
empagus.rates <- extract.ClaDS.rates(maps, 112, 118)
african.rates <- extract.ClaDS.rates(maps, 139, 149)
```

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(ggridges)
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()
```


Polydaedalus_Psammosaurus
Empagusia
Soterosaurus
Varanus_Papusaurus
Euprepiosaurus
Hapturosaurus
Odatria
group
0.0
0.5
1.0
1.5
2.0
sp_rates