

# Monitor Phylogenomics

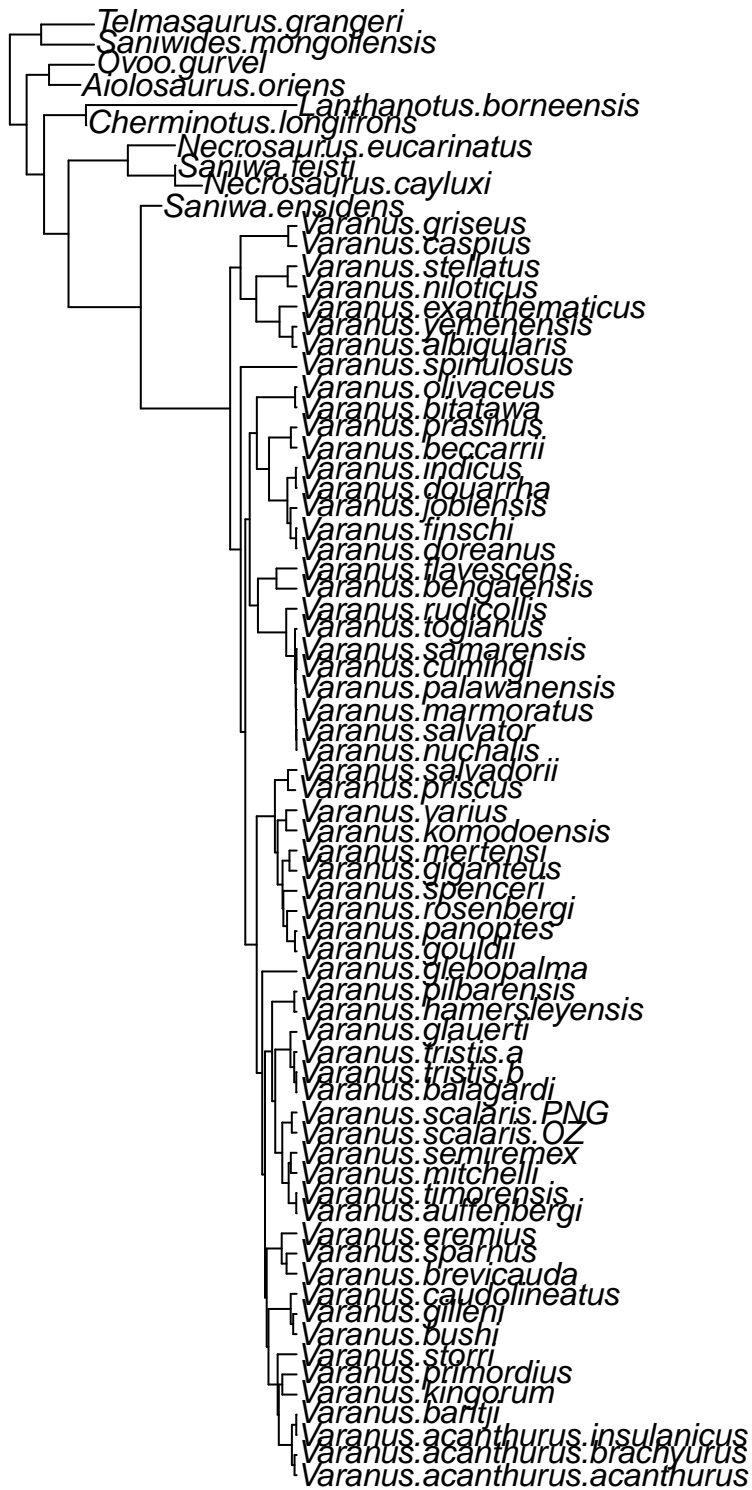
*Ian G. Brennan*

*25/10/2019*

## Look at our data

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

gtree <- read.tree("~/Documents/GitHub/MonitorPhylogenomics/Varanidae_STRICT_HKY_270_con.newick");
plot(gtree)
```



```
alldata <- read.csv("~/Documents/GitHub/MonitorPhylogenomics/All_Size_Data.csv", header=T)
head(alldata)
```

##	SVL	Tail	Name_in_Tree	Location	Group	Habitat
## 1	90.0	NA	Antechinomys.laniger		Marsupial	<NA>
## 2	94.5	NA	Antechinus.agilis		Marsupial	<NA>
## 3	134.5	NA	Antechinus.bellus		Marsupial	<NA>

```
## 4 129.0 NA Antechinus.flavipes Marsupial <NA>
## 5 133.0 NA Antechinus.godmani Marsupial <NA>
## 6 151.0 NA Antechinus.leo Marsupial <NA>
```

Use treplyr to combine the data, then remove any missing

```
goanna <- make.treedata(gtree, alldata)
summary(goanna)
```

```
## A treplyr treedata object
## The dataset contains 5 traits
## Continuous traits: SVL Tail Location
## Discrete traits: Group Habitat
## The following traits have missing values: SVL, Tail, Habitat
## These taxa were dropped from the tree: Antechinomys.laniger, Antechinus.agilis, Antechinus.bellus, A
## These taxa were dropped from the data:
## $phy
##
## Phylogenetic tree with 73 tips and 72 internal nodes.
##
## Tip labels:
## Varanus.acanthurus.acanthurus, Varanus.acanthurus.brachyurus, Varanus.acanthurus.insulanicus, Varan
##
## Rooted; includes branch lengths.
##
## $dat
## # A tibble: 73 x 5
##     SVL Tail Location Group Habitat
##   <dbl> <dbl> <fct>    <fct>    <fct>
## 1 236 370. Australia Varanoidea saxicolous
## 2 236 370. Australia Varanoidea saxicolous
## 3 236 370. Australia Varanoidea saxicolous
## 4 240. 441. Australia Varanoidea saxicolous
## 5 103 228. Australia Varanoidea saxicolous
## 6 120. 165. Australia Varanoidea saxicolous
## 7 144. 217. Australia Varanoidea saxicolous
## 8 114. 159. Australia Varanoidea arboreal
## 9 165. 209. Australia Varanoidea arboreal
## 10 115 136. Australia Varanoidea arboreal
## # ... with 63 more rows
```

```
gf <- filter(goanna, is.na(Tail) == FALSE, is.na(SVL) == FALSE)
```

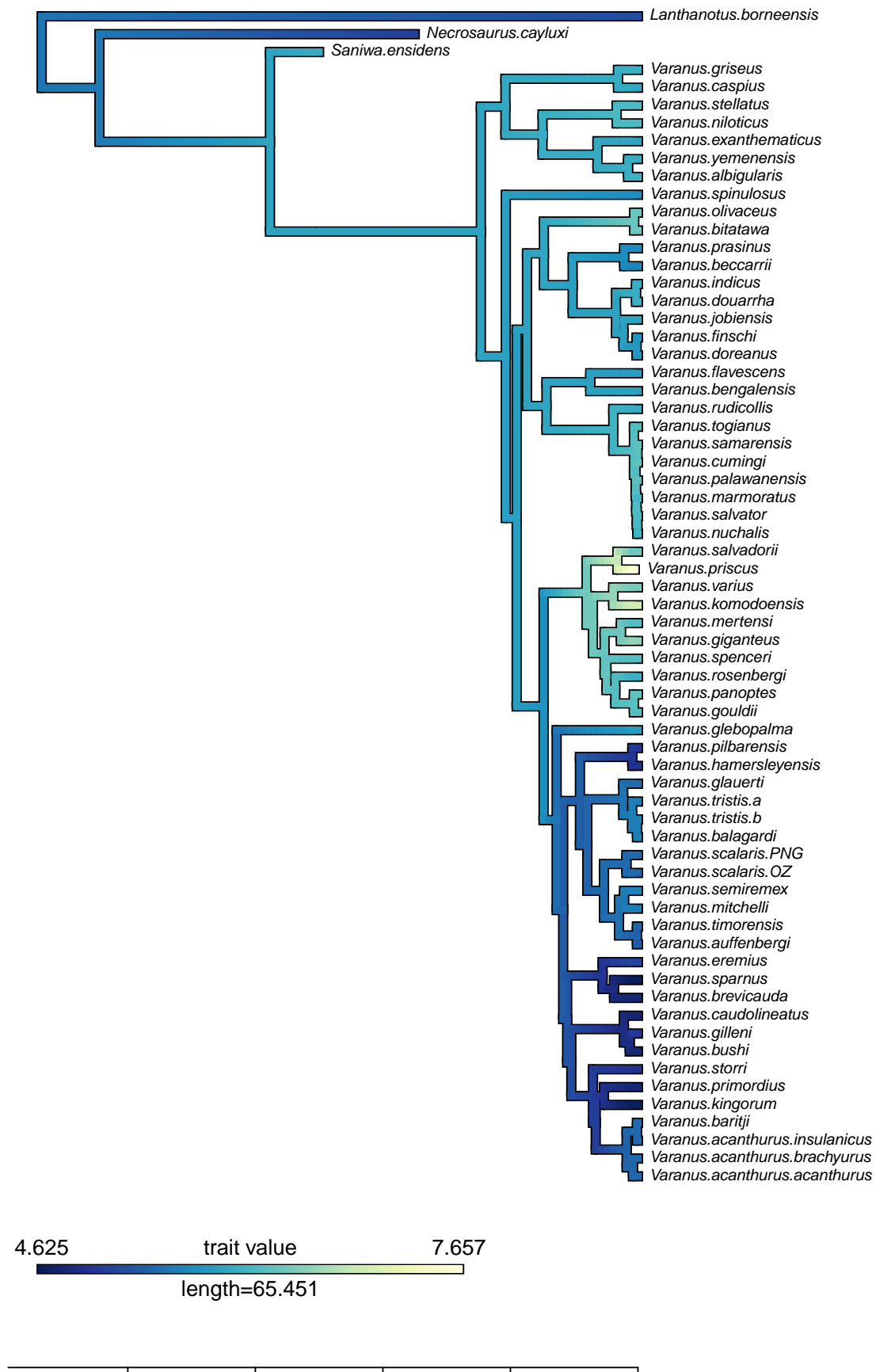
treplyr is great because we can use all the tidyverse terms we already love.

Log-transform the SVL data, and make it a new column

```
gf <- mutate(gf, logSVL = log(gf$dat$SVL))
```

Start visualizing the trait using 'contMap' from 'phytools'

```
obj1 <- contMap(gf$phy, getVector(gf, logSVL), plot=FALSE, outline=F);
n<-length(obj1$cols);
obj1$cols[1:n] <- rev(colorRampPalette(brewer.pal(9, "YlGnBu"))(n));
plot(obj1, legend=0.7*max(nodeHeights(obj1$tree)),
     fsize=c(0.7,0.9), lwd=5, border=F); axisPhylo(1, backward=T)
```



Trim the data just down to Australian goannas

```
oz.g <- filter(gf, Location == "Australia")
summary(oz.g)
```

```
## A treeplyr treedata object
## The dataset contains 6 traits
## Continuous traits: SVL Tail Habitat logSVL
## Discrete traits: Location Group
## The following traits have missing values:
## These taxa were dropped from the tree: Antechinomys.laniger, Antechinus.agilis, Antechinus.bellus, An
## These taxa were dropped from the data:
## $phy
##
## Phylogenetic tree with 35 tips and 34 internal nodes.
##
## Tip labels:
## Varanus.acanthurus.acanthurus, Varanus.acanthurus.brachyurus, Varanus.acanthurus.insulanicus, Varan
##
## Rooted; includes branch lengths.
##
## $dat
## # A tibble: 35 x 6
##       SVL  Tail Location  Group      Habitat    logSVL
##   <dbl> <dbl> <fct>    <fct>    <fct>    <dbl>
## 1  236   370. Australia Varanoidea saxicolous  5.46
## 2  236   370. Australia Varanoidea saxicolous  5.46
## 3  236   370. Australia Varanoidea saxicolous  5.46
## 4  240.  441. Australia Varanoidea saxicolous  5.48
## 5  103   228. Australia Varanoidea saxicolous  4.63
## 6  120.  165. Australia Varanoidea saxicolous  4.78
## 7  144.  217. Australia Varanoidea saxicolous  4.97
## 8  114.  159. Australia Varanoidea arboreal    4.74
## 9  165.  209. Australia Varanoidea arboreal    5.11
## 10 115   136. Australia Varanoidea arboreal    4.74
## # ... with 25 more rows
```

We need to load a bunch of additional packages and custom scripts

```
library(phytools); library(parallel)
library(RPANDA); library(deSolve)
source("/Users/Ian/Google.Drive/R.Analyses/Convenient Scripts/RPANDA_extras.R")
source("/Users/Ian/Google.Drive/R.Analyses/Convenient Scripts/CreateGeoObject_fromSP.R")
source("/Users/Ian/Google.Drive/R.Analyses/Convenient Scripts/Calculate_AICs.R")
source("/Users/Ian/Google.Drive/R.Analyses/Convenient Scripts/plot.distmaps.R")
source("/Users/Ian/Google.Drive/R.Analyses/Convenient Scripts/search.surface.R")
```

The likelihood optimization can be hard given the number of parameters we're estimating, so I've created a function 'search.surface' that uses mclapply to fit the model a number of times with different starting parameters. It starts by creating sets of plausible starting parameters from across the surface, fits them and gives you the output either the best model fit, or all of the model fits. This function/method won't be necessary for simpler models like the BM or OU, and as implemented it won't work on models outside of the RPANDA framework.

```
search.surface <- function(model, n.iter = 10, traits, n.proc = 8,
                           no.S=1, results=c("best", "all"), start.params=NULL)
```

The function lets us control a few things via the commands: \* model—the model of interest, you must have built this already

\* n.iter—the number of model fittings you’d like completed, defaults to 10

\* traits—the input traits for model fitting

\* n.proc—number of processors. this function will fit the model in parallel.

\* no.S—defaults to 1. if your model requires estimating/fitting more than 1 S parameter, say so

\* results—would you like the function to report just the best fitting run, or all the results from each fit attempt

Now we need to build & fit a set of models just to the goanna data to compare with previous hypotheses about how body size variation evolved

Let’s fit a model for goannas based on habitat partitioning (Collar et al. 2011).

This is a multi-OU model with different “adaptive optima” per habitat/niche type.

```
library(OUwie)
source("/Users/Ian/Google.Drive/R.Analyses/Convenient Scripts/make.OUwie.input_Script.R")
goanna.habitat <- make.OUwie.input(data=oz.g$dat, regime=oz.g$dat$Habitat,
                                taxa=oz.g$phy$tip.label, phy=oz.g$phy, trait=oz.g$dat$logSVL)
fitOUM <- OUwie(goanna.habitat$regime.simm, goanna.habitat$combined, model="OUM", simmap.tree=T)
```

Now the standard Brownian Motion (BM) model (Felsenstein, 1985)

```
modelBM <- createModel(oz.g$phy, keyword="BM")
show(modelBM)
fitBM <- fitTipData(modelBM, getVector(oz.g, logSVL), GLSstyle=T)
show(fitBM)
```

Next an Ornstein-Uhlenbeck (OU) model, BM with a single “adaptive optimum”

```
modelOU <- createModel(oz.g$phy, keyword="OU")
show(modelOU)
fitOU <- fitTipData(modelOU, getVector(oz.g, logSVL), GLSstyle=T)
show(fitOU)
```

Build the PM model, which estimates a single S parameter

```
modelPM <- createModel(oz.g$phy, keyword="PM")
show(modelPM)
fitPM <- fitTipData(modelPM, getVector(oz.g, logSVL), GLSstyle=T)
show(fitPM)
```

The geography-informed models below require a processed rase object,

Build the PM+geo model, which estimates a single S parameter, and includes geography

```
modelPMgeo <- createGeoModel(oz.g$phy, goanna.geo.object, keyword="PM+geo")
fitPM_geo <- search.surface(modelPMgeo, n.iter=8, traits=gtraits,
                           n.proc=4, no.S=1, results="best") # this took ~1 min
```

Build the PM OU-less model, which estimates a single S, no alpha, and includes geography

```
modelPMOU_geo <- createGeoModel(goanna.tree, goanna.geo.object, keyword="PMOU+geo")
fitPMOU_geo <- fitTipData(modelPMOU_geo, gtraits, GLSstyle=T)
```

Summarize the model fits

```
multiphy.AIC(prefix="fit", phylo=goanna.tree, models=c("BM", "OU", "PM_geo",
                                                         "PMOU_geo", "OUM"))
```

If you wanted to simulate some data to see what it looks like under the inferred parameters

```
simulateTipData(modelBM, fitBM$inferredParams, method=2)
simulateTipData(modelPMOU_geo, fitPMOU_geo$inferredParams, method=2)
```

We could also quickly check that the inferred params result in simulated data that can recover the correct model:

```
testPMOUg <- simulateTipData(modelPMOU_geo, fitPMOU_geo$inferredParams, method=2)
test_modelBM <- createModel(goanna.tree, keyword="BM")
  test_fitBM <- fitTipData(test_modelBM, testPMOUg, GLSstyle=T)
test_modelPMOU_geo <- createGeoModel(goanna.tree, goanna.geo.object, keyword="PMOU+geo")
  test_fitPMOU_geo <- fitTipData(test_modelPMOU_geo, testPMOUg, GLSstyle=T)
multiply.AIC(prefix="test_fit", goanna.tree, c("BM", "PMOU_geo"))
```

Reminder: you can check the composition of a model using '@', e.g.:

```
modelGMM@aAGamma(3, ssGMM$inferredParams);
modelGMM@aAGamma(3, ssGMM$inferredParams)$A / ssGMM$inferredParams[5]
```