

motmot.2.0

Mark Puttick

2018-07-29

Models Of Trait Macroevolution On Trees (MOTMOT) is an R package that allows for testing of models of trait evolution (Thomas *et al.* 2012).

- Tree transformation models estimated using Maximum likelihood: Brownian motion, Pagel's Lambda, Delta, Kappa, Ornstein-Uhlenbeck (OU), Acceleration-Deceleration (ACDC) and early bursts, psi and multispi, and estimating lambda alongside other models
- Rate heterogeneous models of evolution. Fit models in which the rate of evolution differs in clades selected *a priori* (O'Meara *et al.* 2006; Thomas *et al.* 2006), and models with no *a-priori* shift locations (Thomas *et al.* 2012)
- TimeSlice fit models in which all rates change at a specific time(s) by tested all times or those selected by the user
- Nested Shift mode Fit models in which the ancestral BM rate switches to a 'nested' rate within a monophyletic clade in the phylogeny
- Bayesian estimation of tree transformation models
- Character displacement models of inter-specific competition from Clarke *et al.* (2017)
- Fast estimation of Phylogenetic Generalised Least Squares (PGLS) using independent contrasts

Introduction

First we load motmot.2.0

```
library(motmot.2.0, quietly=T)
```

For these examples we will use anolis data available from motmot. A time-calibrated phylogeny of anolis species ("anolis.tree"), and various trait and biogeographical trait data ("anolis.data")

```
data(anolis.tree)
data(anolis.data)
```

```
names(anolis.data)
```

```
## [1] "Species"      "Island_type"  "ecomorph"     "geo_ecomorph"
## [5] "Female_SVL"   "Male_SVL"
```

```
attach(anolis.data)
anolis.tree
```

```
##
## Phylogenetic tree with 165 tips and 164 internal nodes.
##
## Tip labels:
## A_occultus, A_darlingt, A_monticol, A_bahoruco, A_dolichoc, A_henderso, ...
## Node labels:
## 2, 2, 2, 2, 2, 2, ...
##
## Rooted; includes branch lengths.
```

We will use the continuous trait data: male snout-ventral length 'Male_SVL'. We will construct a matrix of just these data, and check if we have missing data

```
male.length <- matrix(Male_SVL, dimnames=list(rownames(anolis.data)))
any(is.na(male.length[,1]))
```

```
## [1] TRUE
```

We do. So we will remove these data from the male.length data, and log the trait data. This can be done using the function ‘sortTraitData’

```
sortedData <- sortTraitData(anolis.tree, male.length)
phy <- sortedData$phy
male.length <- sortedData$trait
```

Finally, we will ‘prune’ the species from the tree using ‘drop.tip’ from APE. Do our species from the data and tree now match?

```
name.check(phy, male.length)
```

```
## [1] "OK"
```

They do. We can now plot our tree and data using the “traitData.plot” function

```
traitData.plot(y=male.length, phy)
```

We will fit the “tm2” model that allows for clade- and branch-specific changes in rate. This uses the familiar function ‘transformPhylo.ML’. We will fit the models to a subset of these data: including the clade from node 182 only using the APE (Paradis *et al* 2018) function ‘extract.clade’

```
plot(phy, show.tip.label=FALSE, no.margin=TRUE, edge.col="grey20")
nodeLabels(182, 182, bg="black", col="white")
```

```
phy.clade <- extract.clade(phy, 182)
male.length.clade <- as.matrix(male.length[match(phy.clade$tip.label, rownames(male.length)),])
```

Models of trait evolution

We can now test various models of evolution using our trait data.

Brownian motion

To start we will fit a simple Brownian motion model to the data, as the null hypothesis of phylogenetic trait evolution (Cavalli-Sforza and Edwards 1967; Felsenstein 1973; 1985). Brownian motion describes a process in which tip states are a multi-variate normal distribution. On a phylogeny, the multi-variate mean of tip states is equal to the root state estimate, and variance accumulates linearly through time. Until a lineage splits on a tree trait evolution is shared, but following a split individual branches evolve and accumulate variance independently from their shared ancestral value.

```
bm.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="bm")
bm.ml
```

```
## $brownianVariance
##           [,1]
## [1,] 0.00185807
##
## $logLikelihood
## [1] -0.2838554
##
```

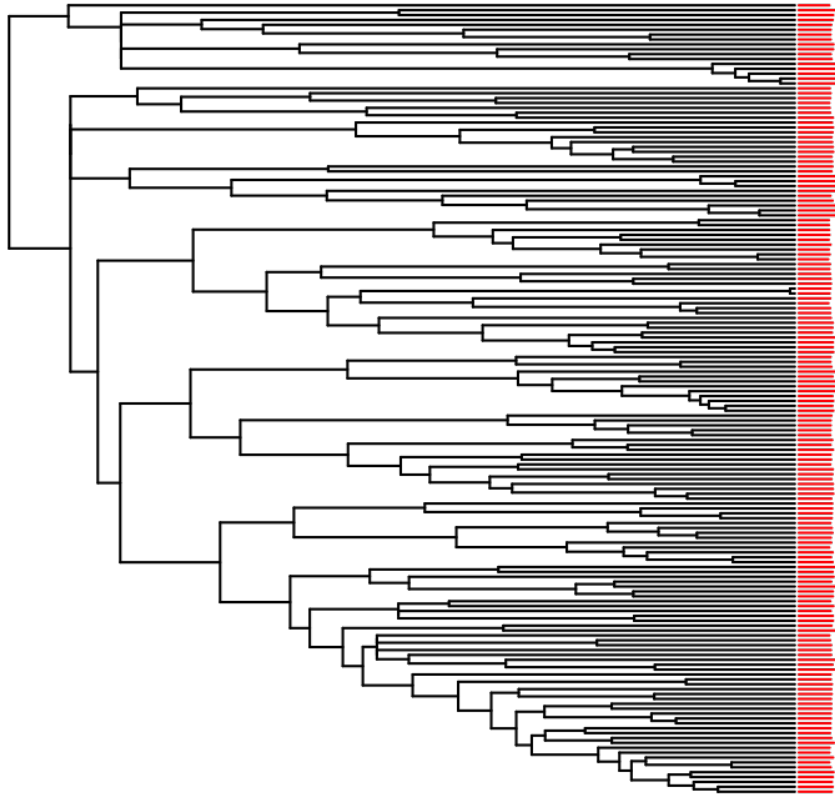


Figure 1: Figure 1. TraitData showing the realtive male snout ventral length at the tips

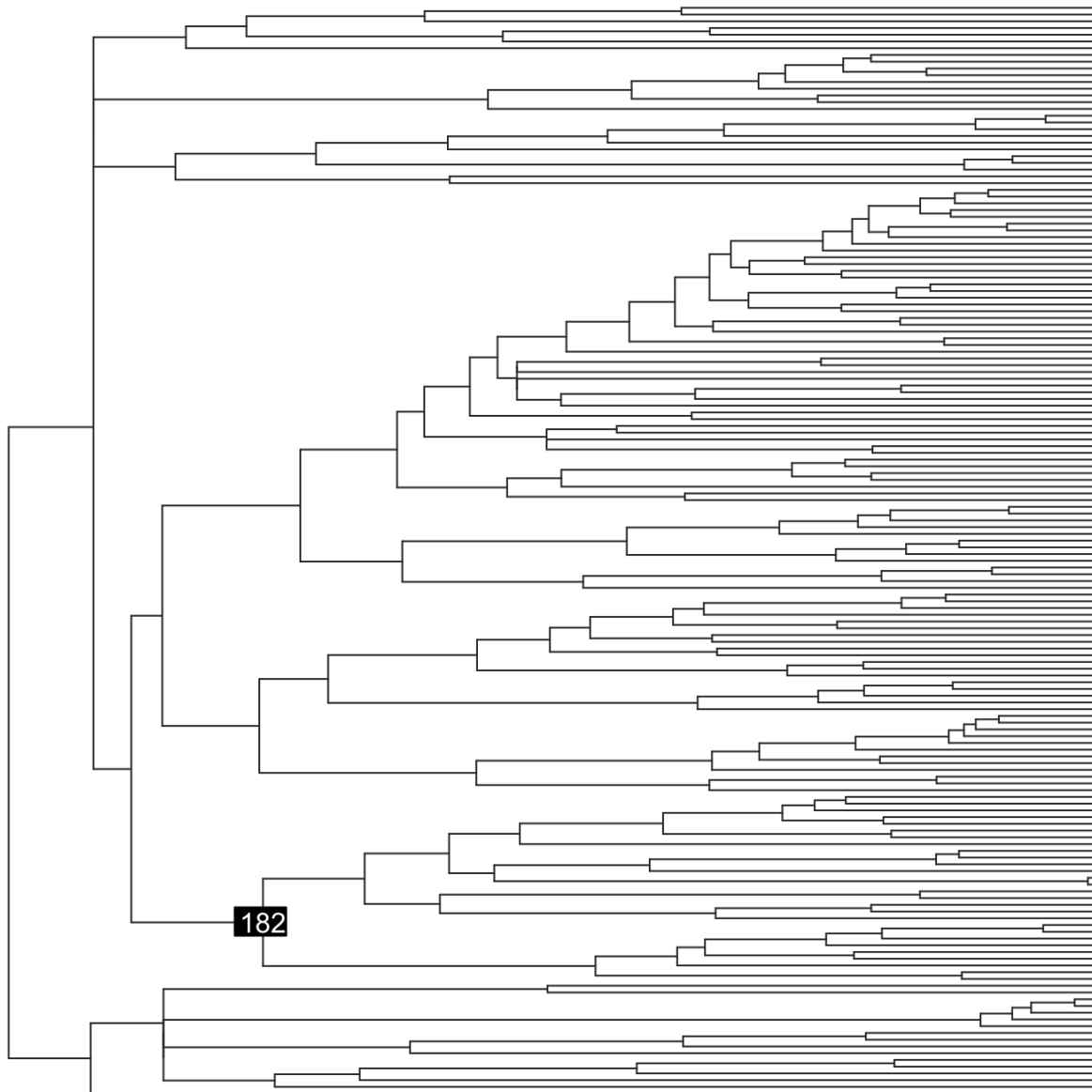


Figure 2: Figure 2. Subset of the anolis tree

```
## $root.state
## [1] 3.849481
##
## $AIC
## [1] 4.567711
##
## $AICc
## [1] 5.047711
```

Pagel's lambda

We can also fit models to test Pagel's lambda (Pagel 1997; 1999). Pagel's lambda is a measure of phylogenetic 'signal' in which the degree to which shared history of taxa has driven trait distributions at tips. In this model, internal branch lengths are changed by the lambda value. If lambda is 1, then branches are equal to a Brownian motion model (high phylogenetic signal), and lower values of lambda indicate less influence of shared history on trait values at the tips. Finally, a value of 0 indicates no phylogenetic influence on trait distributions, and is equivalent to a 'star phylogeny' with no shared branch lengths.

The maximum likelihood of lambda can be estimate in `motmot.2.0`

```
lambda.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="lambda")
lambda.ml
```

```
## $MaximumLikelihood
## [1] 6.522191
##
## $Lambda
##      MLLambda   LowerCI   UpperCI
## [1,] 0.8369989 0.5259423 0.9742338
##
## $brownianVariance
##           [,1]
## [1,] 0.0008245374
##
## $root.state
## [1] 3.853432
##
## $AIC
## [1] -7.044382
##
## $AICc
## [1] -6.044382
```

The maximum likelihood estimate of Pagel's Lambda is equal to 0.83

A new feature in `motmot` allows for plotting of the likelihood profile for the branch-transformation parameter, in this case Pagel's lambda

```
lambda.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="lambda", profilePlot=T)
```

We can now compare the fit of the BM and Lambda models. Lambda has higher likelihood, but it also has more parameters (the root state and sigma squared are shared in both models, lambda also estimates the lambda parameter). We can test whether this is a significant improvement. First we will use the chi-squared distribution. The models differ in one degree of freedom: BM has 2 parameters (brownian variance, root state) and lambda has those two parameters plus the value of lambda, so 3 parameters. We can use the `stats` function `pchisq` to obtain a p value, and see that lambda is indeed a superior fit to these data

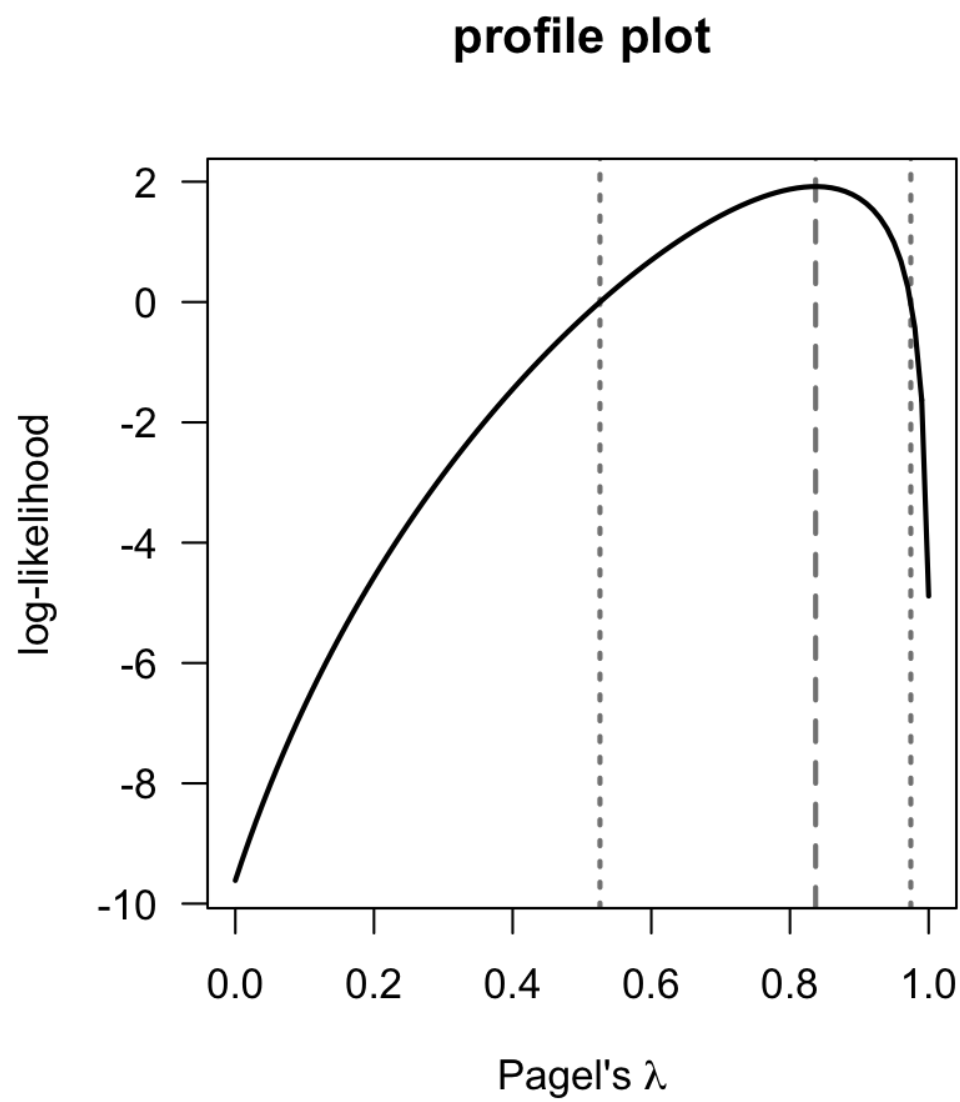


Figure 3: Figure 3. Profile plot of ML estimation for Pagel's lambda

```
p.value <- 1 - pchisq(lambda.ml$MaximumLikelihood - bm.ml$logLikelihood, 1)
p.value
```

```
## [1] 0.009084969
```

Additionally there is a large Akaike Information Criterion (AICc) difference between the two models: BM has a higher AICc compared to Lambda. The difference (11.09) is >4 which is traditionally seen as indication of a superior fit (Burnham and Anderson 2003).

```
bm.ml$AICc - lambda.ml$AICc
```

```
## [1] 11.09209
```

The parameters, brownian variance, root state, Maximum likelihoods, AIC, and AICc can be obtained for a number of models in *motmot*.

Delta

Delta indicates a slow or increase in the rate of trait evolution through time (Pagel 1997; 1999); a value of 1 is equivalent to Brownian motion, < 1 indicates a slow-down, and > 1 is difficult to interpret (greater change near the present). Here we find a MLE of 2.23 but the CI spans < 1 to > 4

```
delta.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="delta", profilePlot=T)
```

```
delta.ml
```

```
## $MaximumLikelihood
## [1] 1.179784
##
## $Delta
##      MLDelta   LowerCI UpperCI
## [1,] 2.231467 0.8477548 4.66072
##
## $brownianVariance
##           [,1]
## [1,] 6.013921e-06
##
## $root.state
## [1] 3.8843
##
## $AIC
## [1] 3.640431
##
## $AICc
## [1] 4.640431
```

Kappa

Kappa is used as a measure of punctuated evolution and spans values of 0-1 (Pagel 1997:1999). 1 is equivalent to BM, and 0 indicates trait change occurs at events of speciation. Here there is evidence of punctuated evolution (the warning message simply tells out the CI falls outside the parameter bounds - in this case below zero).

```
kappa.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="kappa", profilePlot=T)
```

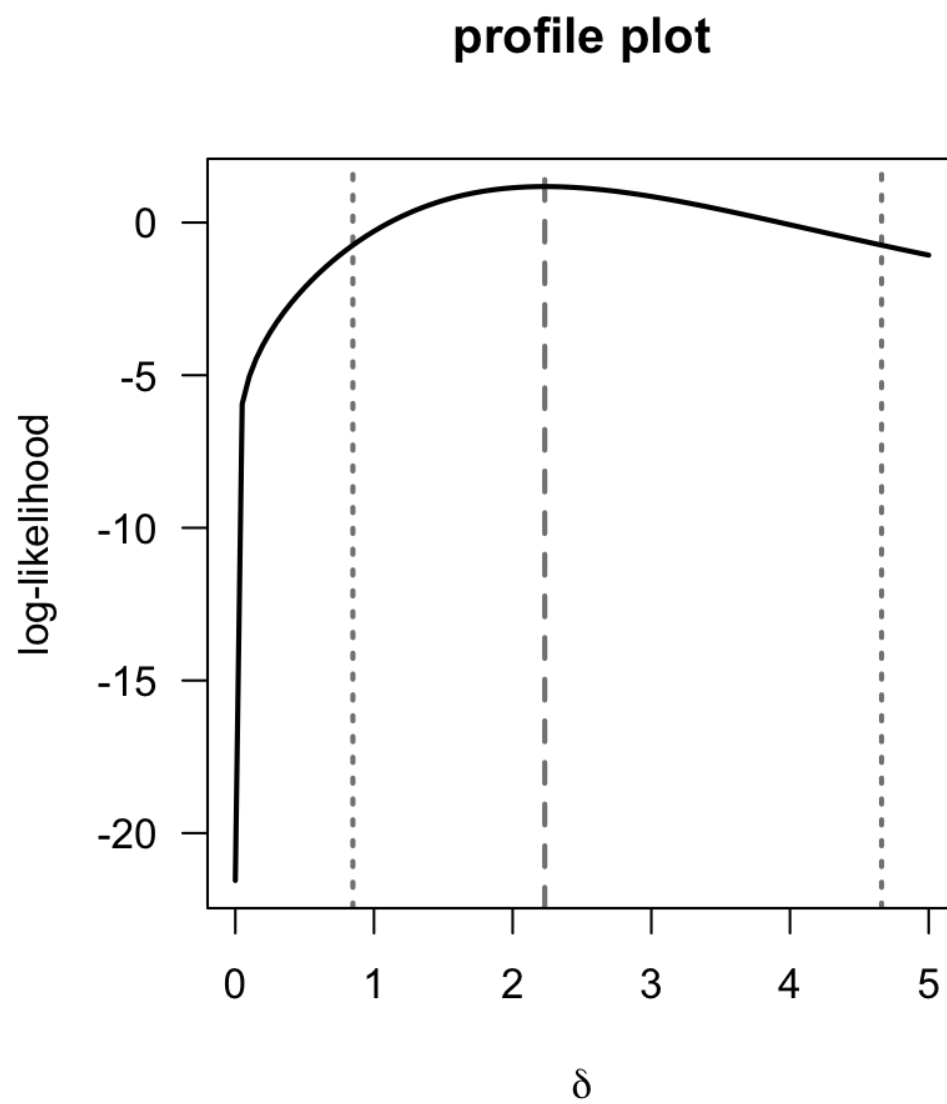


Figure 4: Figure 4. Profile plot to estimate delta


```
## Warning in lower.function.warning(): Confidence limits fall outside
## parameter bounds - consider changing lowerBound
```

Ornstein-Uhlenbeck

The OU model allows for modelling of attraction to a optimum value, alpha (Hansen 1997; Butler and King 2004). This model again is similar to the Brownian motion model, but models the strength of attraction to alpha. The OU model can be difficult to interpret and care is advised in its use (Cooper *et al.* 2016).

In motmot.2.0, as with most implements of the phylogenetic OU model, the value of attraction parameter is equal to the ancestral trait estimate.

```
ou.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="OU", profilePlot=T)
```

```
ou.ml
```

```
## $MaximumLikelihood
## [1] 1.743448
##
## $Alpha
##      MLAlpha      LowerCI  UpperCI
## [1,] 0.01692855 0.0004499246 0.039123
##
## $brownianVariance
##      [,1]
## [1,] 0.002823518
##
## $root.state
## [1] 3.876694
##
## $AIC
## [1] 2.513103
##
## $AICc
## [1] 3.513103
```

The value of alpha is higher than zero, but very small (0.01692855). So the model is not equivalent to Brownian motion but there is little evidence from AICc that the model is an improvement, and the likelihood ratio test show a non-significant improvement

```
p.value <- 1 - pchisq(ou.ml$MaximumLikelihood - bm.ml$logLikelihood, 1)
p.value
```

```
## [1] 0.1544945
```

```
bm.ml$AICc - ou.ml$AICc
```

```
## [1] 1.534607
```

ACDC and Early Burst

A new addition to MOTMOT is the ACDC model (Blomberg *et al.* 2003). This model allows for exponential changes in the rate of evolution in the history of a clade.

```
acdc.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="ACDC", profilePlot=T)
```

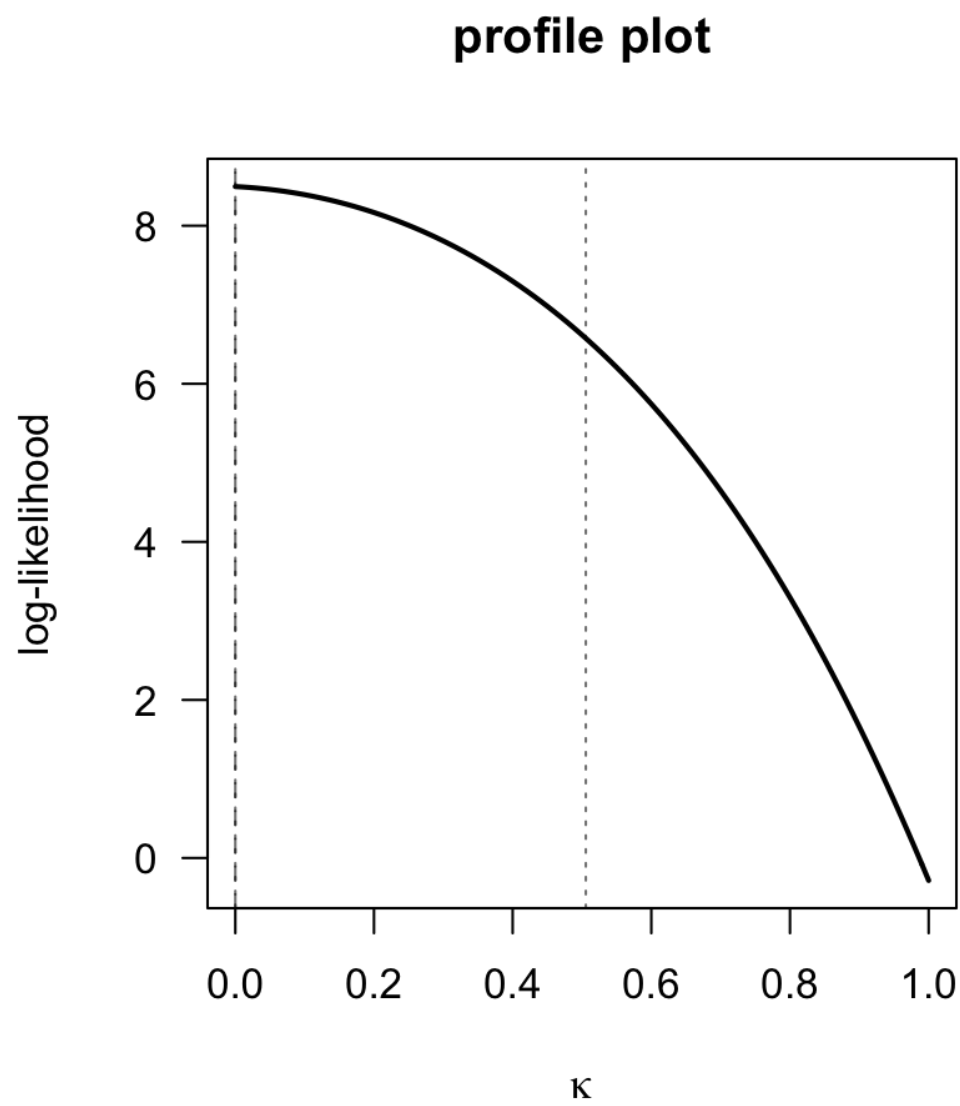


Figure 5: Figure 5. Profile plot to estimate kappa

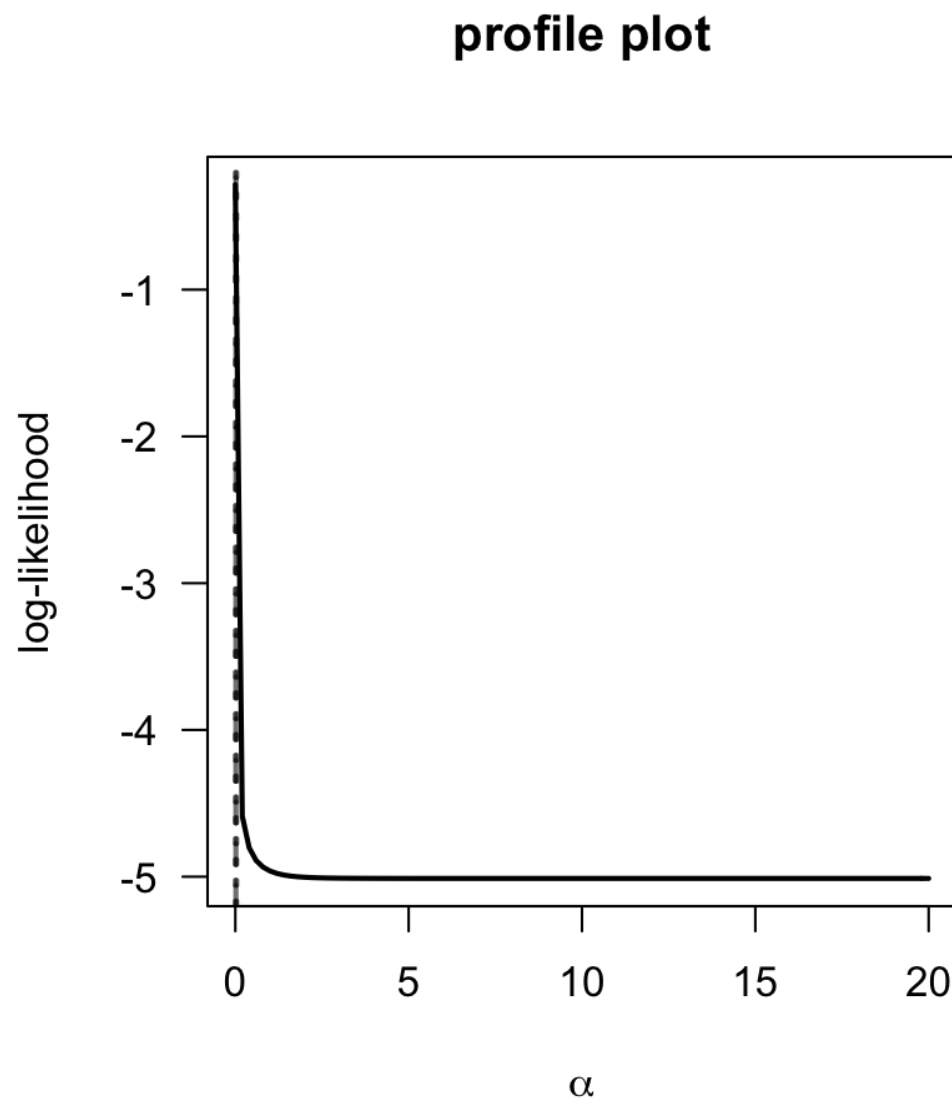


Figure 6: Figure 6. Profile plot to estimate alpha

```
## Warning in upper.function.warning(): Confidence limits fall outside
## parameter bounds - consider changing upperBound
```

```
acdc.ml
```

```
## $MaximumLikelihood
## [1] 1.743448
##
## $ACDC
##           MLacdc      LowerCI      UpperCI
## [1,] 0.03385987 0.0008792981 0.04246516
##
## $brownianVariance
##           [,1]
## [1,] 0.000259074
##
## $root.state
## [1] 3.876696
##
## $AIC
## [1] 2.513103
##
## $AICc
## [1] 3.513103
```

There is little evidence here of exponential decreases or increases in the rate of trait evolution - the acdc exponential parameter is close to 0 (0.034). We can see this is not a significant improvement on BM

```
p.value.2 <- 1 - pchisq(acdc.ml$MaximumLikelihood - bm.ml$logLikelihood , 1)
p.value.2
```

```
## [1] 0.1544945
```

As an example, here we constrain the ‘upperBound’ to < 0, this is equivalent to the Early Burst model (Harmon *et al.* 2010) fit in geiger

```
transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="ACDC", profilePlot=FALSE, upperBound=-1e-6
```

```
## $MaximumLikelihood
## [1] -0.2839779
##
## $ACDC
##           MLacdc      LowerCI      UpperCI
## [1,] -1e-06 -0.01322546 -1e-06
##
## $brownianVariance
##           [,1]
## [1,] 0.001858183
##
## $root.state
## [1] 3.849481
##
## $AIC
## [1] 6.567956
##
## $AICc
## [1] 7.567956
```

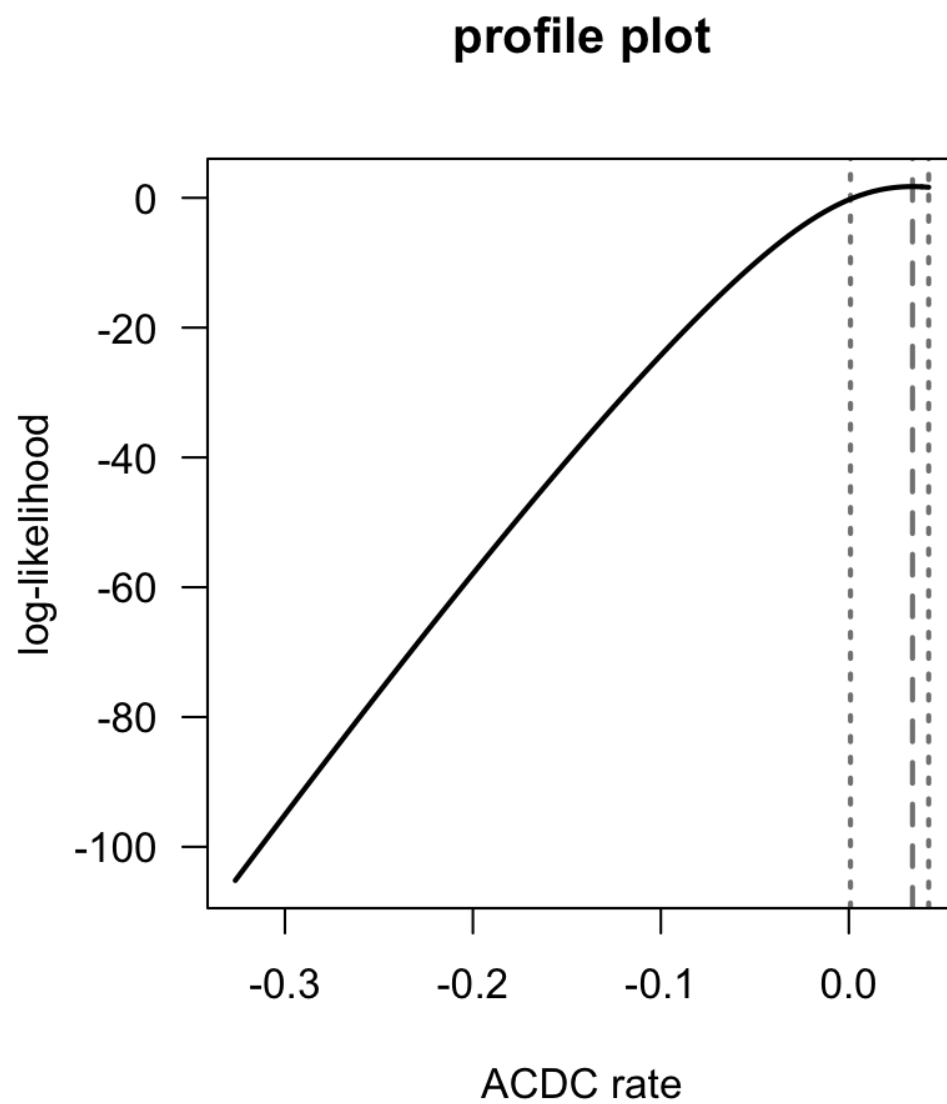


Figure 7: Figure 7. Profile plot to estimate the ACDC parameter

The estimate of $-1e-6$ for the exponential decrease parameter, means the model is effectively equivalent to Brownian motion

psi and multispi

The parameter psi is similar to the parameter kappa in that it measures the relative contribution of speciational (~punctuated) and gradual evolution to trait change (Ingram 2011; Ingram *et al.* 2016). The parameter psi is based upon measures of evolution over time and at speciation, and can also account for ‘hidden’ nodes not seen in the input phylogeny. The parameter psi measures the proportion of speciational change and background time, so the estimation for psi between 0 (Brownian motion) and 1 (indicating equal branch lengths, ~speciational change).

In motmot we can fit a simple psi model using the input tree.

```
psi.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="psi", profilePlot=T)
```

```
## Warning in upper.function.warning(): Confidence limits fall outside
## parameter bounds - consider changing upperBound
```

```
psi.ml

## $MaximumLikelihood
## [1] 8.495569
##
## $psi
##      MLpsi   LowerCI UpperCI
## [1,]      1 0.3160581      1
##
## $brownianVariance
##              [,1]
## [1,] 0.0008018518
##
## $root.state
## [1] 3.761269
##
## $AIC
## [1] -10.99114
##
## $AICc
## [1] -9.991139
```

This indicates strong support for the psi model.

```
p.value.psi <- 1 - pchisq(acdc.ml$MaximumLikelihood - bm.ml$logLikelihood , 1)
p.value.psi
```

```
## [1] 0.1544945
```

And it is significant improvement on Brownian motion

We could also get a potentially more accurate of speciation rates by using the full tree, rather than the pruned tree. If death rates are over 0, then the estimates will differ from the simple model above.

```
transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="psi", profilePlot=FALSE, hiddenSpeciation=
```

```
## Warning in upper.function.warning(): Confidence limits fall outside
## parameter bounds - consider changing upperBound
```

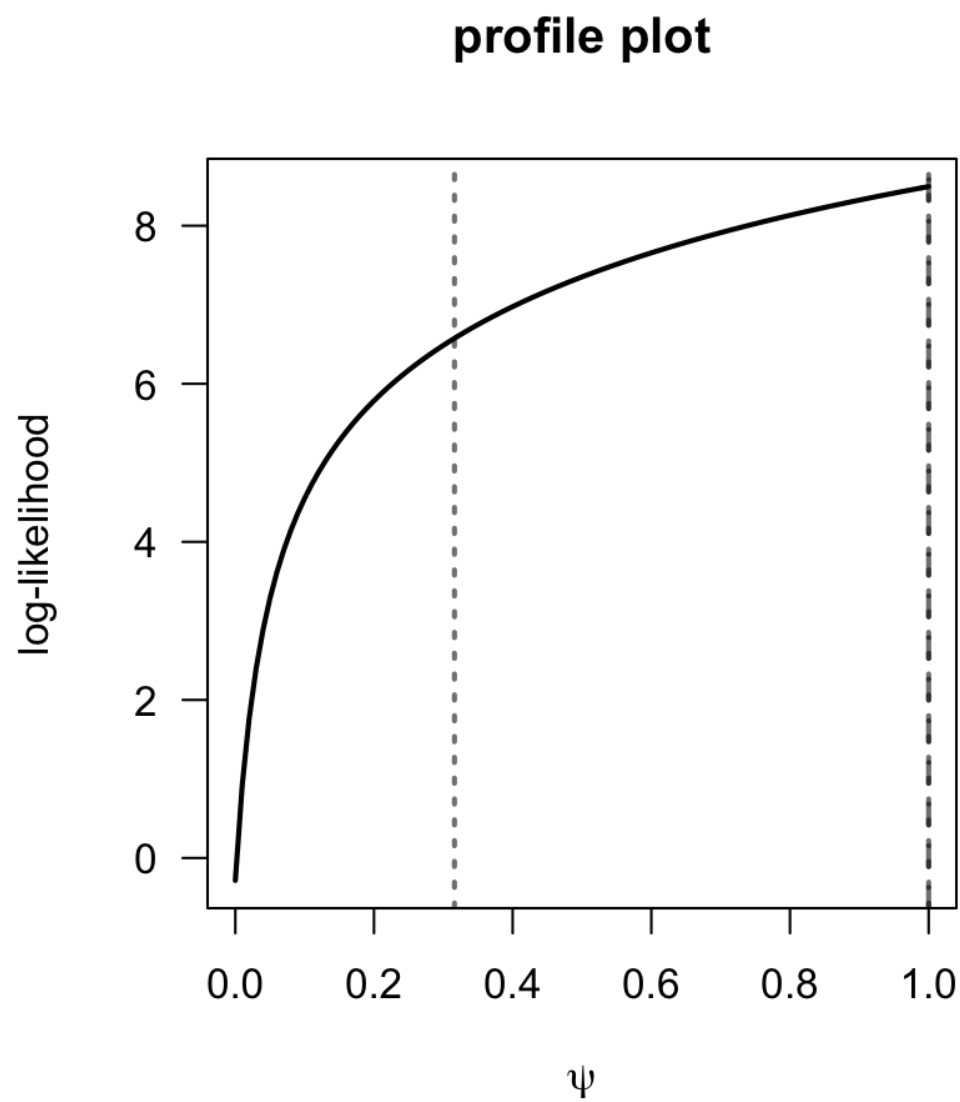


Figure 8: Figure 8. Profile plot to estimate the psi parameter

```
## $MaximumLikelihood
## [1] 8.495569
##
## $psi
##      MLpsi    LowerCI UpperCI
## [1,]      1 0.3160581      1
##
## $brownianVariance
##              [,1]
## [1,] 0.0008018518
##
## $root.state
## [1] 3.761269
##
## $AIC
## [1] -10.99114
##
## $AICc
## [1] -9.991139
```

In this case, there is no difference in the estimates as death rates are equal to 0

We can also apply multipsi model in which different regions of the tree have different estimates of the parameter psi

```
plot(phy.clade, no.margin=TRUE, cex=0.8)
two.clade.labels <- c(rep("a", 17), rep("b", 37))
edgelabels(two.clade.labels, col=c(rep("blue", 17), rep("red", 37)), bg="white")
```

We can now fit the multipsi model with these data

```
transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="multipsi", branchLabels=c(rep("a", 17), rep("b", 37)))
```

```
## $MaximumLikelihood
## [1] 8.495569
##
## $psi
##      MLpsi    LowerCI UpperCI
## a      1 0.04812271      NA
## b      1 0.20955325      NA
##
## $brownianVariance
##              [,1]
## [1,] 0.0008018518
##
## $root.state
## [1] 3.761269
##
## $AIC
## [1] -8.991139
##
## $AICc
## [1] -7.252008
```

In this model, the estimate of psi does not differ between the two regions of the tree

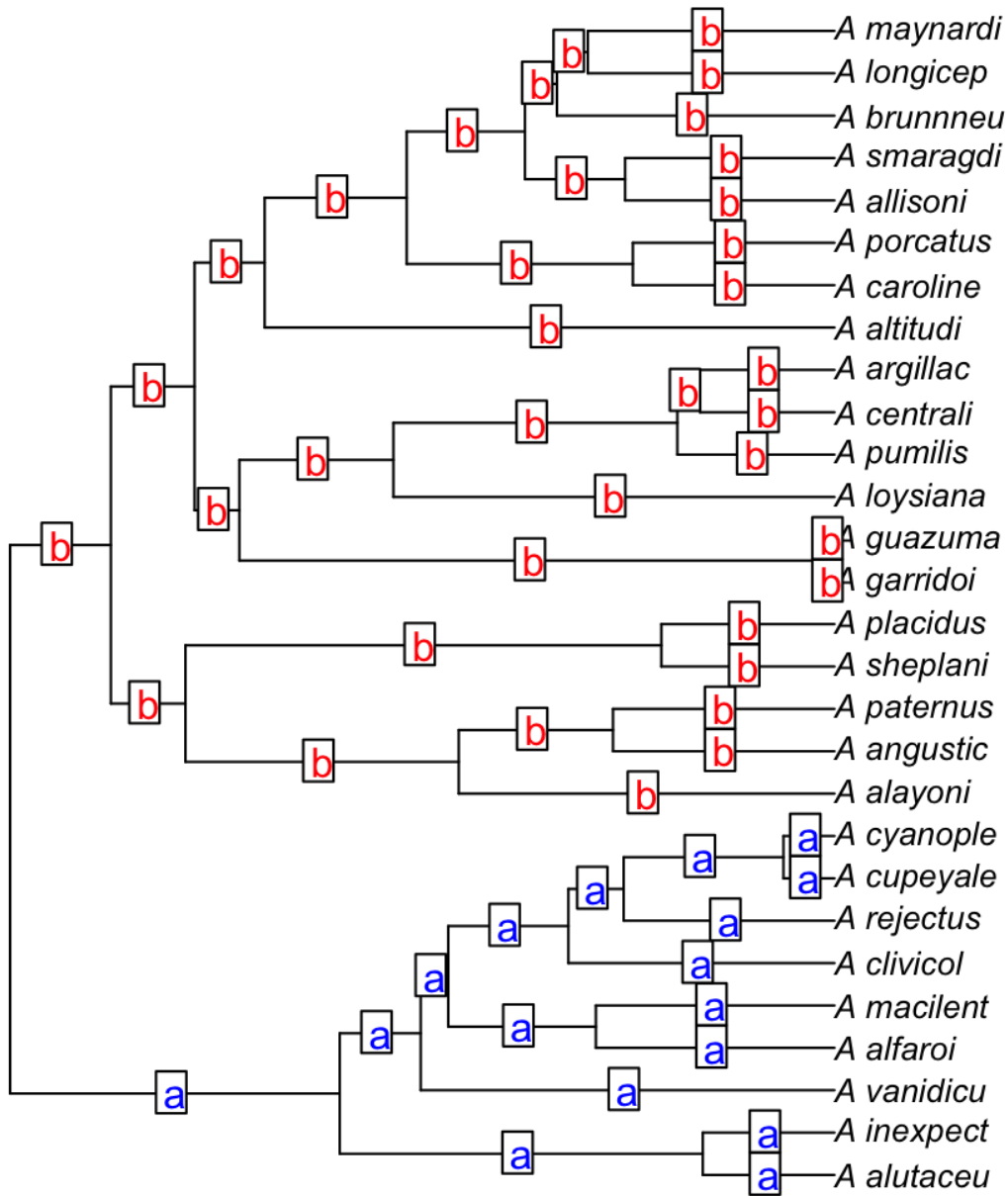


Figure 9: Figure 9. Two clades used in the multipsi model

Estimate lambda alongside models

One way to deal with ‘noisy’ data is to estimate Pagel’s lambda alongside a parameter of interest. In *motmot*, lambda can be estimated alongside the delta, kappa, OU, psi, and ACDC models. Here we look at example using ACDC. The model is fit with same function. ‘transformPhylo.ML’, but with the argument ‘lambdaEst’ set to TRUE

```
acdc.ml.lambda <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="ACDC", lambdaEst=T)
# original ACDC model
acdc.ml
```

```
## $MaximumLikelihood
## [1] 1.743448
##
## $ACDC
##           MLacdc      LowerCI      UpperCI
## [1,] 0.03385987 0.0008792981 0.04246516
##
## $brownianVariance
##           [,1]
## [1,] 0.000259074
##
## $root.state
## [1] 3.876696
##
## $AIC
## [1] 2.513103
##
## $AICc
## [1] 3.513103
# ACDC model plus lambda
acdc.ml.lambda
```

```
## $MaximumLikelihood
## [1] 7.376867
##
## $ACDC
##           MLacdc      LowerCI      UpperCI
## [1,] -0.182928 -0.3243635 -0.08288833
##
## $brownianVariance
##           [,1]
## [1,] 0.01272343
##
## $root.state
## [1] 3.836039
##
## $lambda
## [1] 0.1389153
##
## $AIC
## [1] -4.753735
##
## $AICc
## [1] -2.026462
```

We can see λ is < 1 , and this has affected the parameter estimation. The improvement in the model fit is significant compared to the ACDC model without λ , and the null BM model

```
# p value of the ACDC and ACDC+lambda models. No significant improvement
1 - pchisq(acdc.ml.lambda$MaximumLikelihood - acdc.ml$MaximumLikelihood , df=1)

## [1] 0.01762123

# p value of the BM and ACDC+lambda model comparison. No significant improvement
1 - pchisq(acdc.ml.lambda$MaximumLikelihood - bm.ml$logLikelihood, df=2)

## [1] 0.02170177
```

Rate heterogeneous models of evolution

rate heterogeneity selected *a priori*

MOTMOT can test models of evolution in which pre-defined clades can vary in the rate of evolution. Here we fit a model in which the nodes descending from nodes 32 and 49 have a separate rate of evolution. We can visualise these nodes on the phylogeny

```
plot(phy.clade, show.tip.label=F, no.margin=T, edge.col="grey20")
nodeLabels(c(32, 49), c(32, 49), bg="black", col="white")
```

We then fit the motmot model, again using the function transformPhylo.ML. We use the argument “model=clade”. This fits the non-censored model of O’Meara *et al.* (2006).

```
cladeRate.ml <- transformPhylo.ML(phy=phy.clade, y=male.length.clade, model="clade", nodeIDs=c(32, 49))
cladeRate.ml

## $Rates
##      node      MLRate   LowerCI  UpperCI
## [1,]    32 0.8138083 0.2632949 3.263621
## [2,]    49 0.6819058 0.1896342 2.952355
##
## $MaximumLikelihood
## [1] -0.1462706
##
## $brownianVariance
##           [,1]
## [1,] 0.002143263
##
## $root.state
## [1] 3.870488
##
## $AIC
## [1] 8.292541
##
## $AICc
## [1] 10.03167
```

These results indicate that the two clades tend to have a lower rate of evolution compared to the background rate. However, the CIs indicate these decreases may not be robust

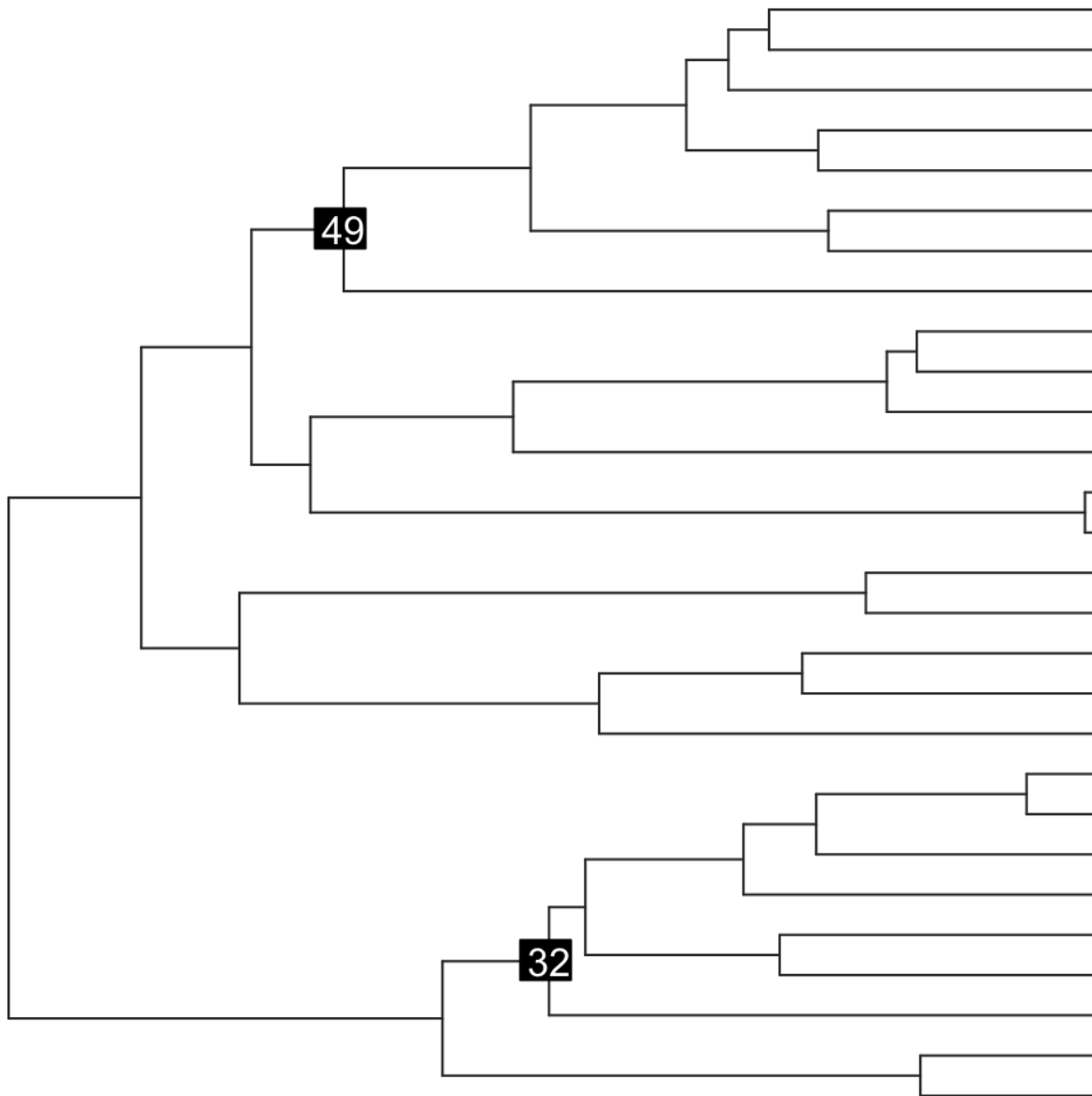


Figure 10: Figure 10. Lineages with different rates of evolution

rate heterogeneity with no *a priori* information

We can also fit rate heterogeneous models without specifying where we expect shifts on the tree. We can use the arguments “model=“tm1” and “model=“tm2”; these models fit ‘traitMedusa’ models in which all nodes are tested for rate increases or decreases. It is possible to exclude small nodes using the argument ‘minCladeSize’. As well as allowing clade differences in rate, the “tm2” also allows for branch-based increases or decreases in rate.

We can now fit the ‘tm2’ algorithm. The output shows the log-likelihood, AIC, AICc, rate type (branch of clade), for the best-fitting model at each stage. This starts with the BM model, and then one shift model, two shift model, etc.,

```
# not run
# tm1.ml <- transformPhylo.ML(y=male.length.clade, phy=phy.clade, model="tm1", minCladeSize=2, nSplits=
# trait.medusa.tm1.summary <- traitMedusaSummary(tm1.ml, cutoff=2, AICc=T)
tm2.ml <- transformPhylo.ML(y=male.length.clade, phy=phy.clade, model="tm2", minCladeSize=5, nSplits=2)

##
## BM model
##      node shiftPos      lnL n.params      AIC      AICc
## BM      0          1 -0.2838554      2 4.567711 5.047711
##
## Shift 1
##      node shiftPos      lnL n.params      AIC      AICc      rate.1
## shift.1  39      clade 3.042343      3 -0.08468691 0.9153131 0.09148634
##
## Shift 2
##      node shiftPos      lnL n.params      AIC      AICc      rate.1
## shift.2  44      clade 4.746778      5 0.5064434 3.233716 0.1408067
##      rate.2
## shift.2 3.158572
```

We can now summarise the results of these data using ‘traitMedusaSummary’ and plotting the shifts on the phylogeny using ‘plotPhylo.motmot’. These results show a decrease at node 39 that we can visualise on the phylogeny.

```
trait.medusa.tm2.summary <- traitMedusaSummary(tm2.ml, cutoff=2, AICc=T)
trait.medusa.tm2.summary

## $ModelFit
##      lnL n.params      AIC      AICc
## shift.1 3.042343      3 -0.08468691 0.9153131
##
## $Rates
##      node shiftPos      MLRate      LowerCI      UpperCI
## 1    39      clade 0.0914863409977509 0.0260312941371971 0.500387096823083
##
## $OptimalTree
##
## Phylogenetic tree with 28 tips and 27 internal nodes.
##
## Tip labels:
## A_alutaceu, A_inexpect, A_vanidicu, A_alfaroi, A_macilent, A_clivicol, ...
## Node labels:
## 2, 2, 2, 2, 2, 2, ...
##
```

```
## Rooted; includes branch lengths.
```

```
colour_motmot <- plotPhylo.motmot(phy=phy.clade, traitMedusaObject=trait.medusa.tm2.summary, reconType =
```

Thomas and Freckleton (2012) showed the tm2 algorithm has a high type-one error rate. One way to ameliorate this is to estimate the level a one shift is supported when we know BM is the true model. For example, we could simulate 1000 BM datasets on the tree, estimate a single shift using the tm2 algorithm, and calculating the difference between the AICcs for each BM and one shift model. We can then use this difference to estimate the AICc ‘penalty’ that is needed to reduce the tm2 type-one error rate to 0.05. We could use this penalty in the ‘cutoff’ argument of the traitMedusaSummary argument.

This is shown but not run in the code below

```
# not run
# sim.bm <- transformPhylo.sim(phy=phy.clade, n=1000, model="bm")
# aic.cut.off <- apply(sim.bm, 2, function(x) {
#   # bm.test <- transformPhylo.ML(y=as.matrix(x), phy=phy.clade, model="tm2", minCladeSize=2, nSplits=
#   # bm.test[[1]][,"AICc"]
#   # })
# tm2.cut.off <- quantile(aic.cut.off[1,] - aic.cut.off[2,], 0.95)
```

Time-slice model

A new addition to motmot is a Maximum likelihood model that allows for heterogeneous rates in different times of evolution. These models are separate from the models that allow for heterogeneous rates among lineages, as modelled by the ‘traitMedusa’ algorithms.

The ‘timeSlice’ model is implemented using the ‘transformPhylo.ML’ function, using the argument model = ‘timeSlice’. The function allows for two separate models of evolution. In one, it is possible to test shifts in evolution at times selected *a priori*. Alternatively, the fit of models can be tested at a range of different times, and the function will return the best-fitting model

First we will test for a shift in the rate of evolution 10 million years ago.

```
timeSlice.10.ml <- transformPhylo.ML(y=male.length.clade, phy=phy.clade, model="timeSlice", splitTime=c
```

```
## [1] "BM model"
##          lnL          AIC          AICc      sigma.sq      anc.state
## -0.28385540  4.56771081  5.04771081  0.00185807  3.84948140
## [1] "shiftModel"
##          lnL          AIC          AICc      sigma.sq      anc.state
##  2.946487446  2.107025107  3.846155542  0.001006388  3.860015287
##          rate1          rate2      time.split1
##  0.692072848  2.944767886  10.000000000
```

We can use the function ‘timeSliceSummary’ to plot and summarise the results. The output summarises the best model according to AICc values. This function automatically plots the original tree showing the location of shift(s), and the colours show the relative rates in each time slice. The second plot below shows the same tree and colours, but with the branch lengths scaled to the ML optimised rates

```
outputSummary <- timeSliceSummary(timeSlice.10.ml, cutoff=0.001, cex.tip=0.2, phylo.width=2, colour.ram
```

We can also see other summarise information, such as the CI for each rate estimate.

```
outputSummary$RatesCI
```

```
##          rates          LCI          UCI
## rate1 0.6920728 0.1873338  2.115629
```

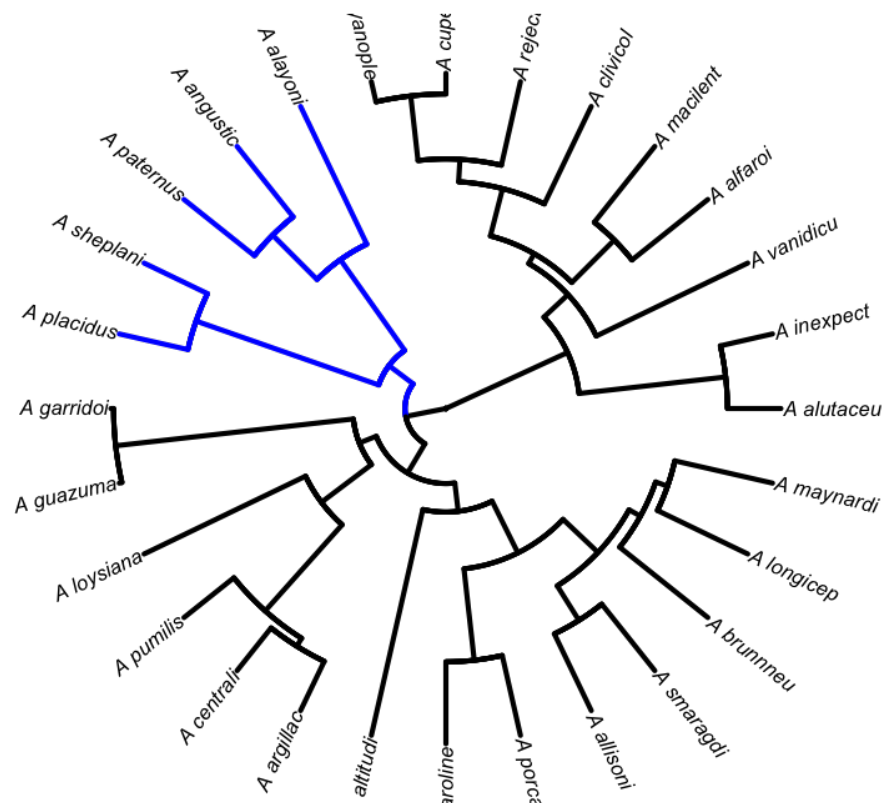


Figure 11: Figure 11. The subset of the tree

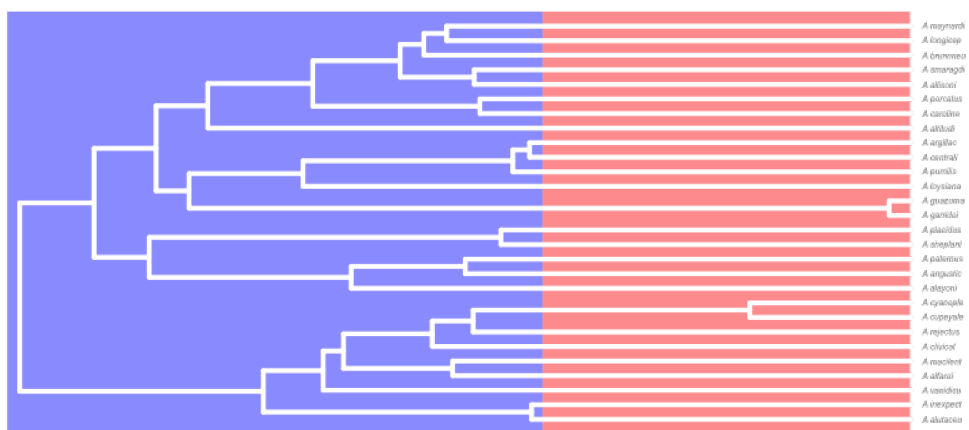


Figure 12: Figure 12. TimeSlice plot with a split at 10 Ma


```
## rate2 2.9447679 0.9633096 10.877404
```

Rather than testing the overall fit of each model, we can fit models to all times. The function automatically tests for all 1 Ma shifts between the age of the tree - 10 Ma, and the present + 10 Ma. We can specify a number of shifts we would like to test for. Here we will test for up to 3 shifts. The model will test one shift, save it, search for a second, save those two, etc...

Here will modify the boundary age argument so all split times are tested between 62-8 Myrs, using the 'boundaryAge' argument. As we are not tested set times we need to set the number of splits to test using 'nSplits' - we will allow up to 2 splits

```
timeSlice.ml <- transformPhylo.ML(y=male.length.clade, phy=phy.clade, model="timeSlice", nSplits=2, bound
```

```
## [1] BM model
##          lnL          AIC          AICc      sigma.sq    anc.state
## -0.28385540  4.56771081  5.04771081  0.00185807  3.84948140
## [1] shift 1
##          lnL          AIC          AICc      sigma.sq    anc.state      rates1
## 3.401719137 1.196561726 2.935692160 0.001008072 3.859633059 0.675082529
##          rates2 time.split1
## 3.156494908 8.545642000
## [1] shift 2
##          lnL          AIC          AICc      sigma.sq    anc.state
## 3.825827e+00 2.348347e+00 5.075620e+00 1.067401e-04 3.857196e+00
##          rates1      rates2      rates3 time.split1 time.split2
## 7.002769e+00 1.000000e-08 3.010568e+01 8.545642e+00 1.354564e+01
```

And summarise the results. We can selected the cutoff AICc improvement needed to justify selecting the next model. Here we use the arbitrary cut-off value of 1. We could test this formally by estimating the correct AICc value needed to reduced type-error > 5% by using BM simulated data (an example using the tm2 is shown above)

```
outputSummary <- timeSliceSummary(timeSlice.ml, cutoff=1, cex.tip=0.2, phylo.width=2, colour.ramp=c("bl
```

Nested models of evolution

We can also tested models of nested evolution in which an ancestral model of BM evolution changes to a alternative model (EB, OU, kappa, delta, psi) within the phylogeny (Puttick 2018).

Here we can show an example of BM -> OU and BM -> ACDC at node 44 of the phylogeny. However, neither of these is significantly better than BM

```
bm.model <- transformPhylo.ML(male.length.clade, phy=phy.clade, model="bm")
nested.acdc <- transformPhylo.ML(male.length.clade, phy=phy.clade, model="ACDC", nodeIDs=c(44))
```

```
## Warning in upper.function.warning(): Confidence limits fall outside
## parameter bounds - consider changing upperBound
```

```
nested.ou <- transformPhylo.ML(male.length.clade, phy=phy.clade, model="OU", nodeIDs=c(44))
```

```
## Warning in lower.function.warning(): Confidence limits fall outside
## parameter bounds - consider changing lowerBound
```

```
1 - pchisq(nested.acdc$MaximumLikelihood - bm.model$logLikelihood, 1)
```

```
## [1] 0.05740847
```



```
1 - pchisq(nested.ou$MaximumLikelihood - bm.model$logLikelihood, 1)
```

```
## [1] 0.361424
```

Bayesian estimation of tree transformation models

The function ‘transformPhylo.MCMC’ allows for the estimation of model parameters using Bayesian statistics. Models of lambda, delta, kappa, OU, ACDC, and psi can currently be modelled using transformPhylo.MCMC

The model allows for a pre-optimisation step. The model we test 30 (default) different deviations for the acceptance proposal distribution in order for the model to achieve an acceptance of around 0.44. This is done by default in the model but can be turned off by setting ‘opt.accept.rate=FALSE’

We will run an MCMC chain of 1000 generations to estimate Pagel’s lambda and discarding the first 10% (‘200 generations (‘burn.in = 0.1’). All the models use a ‘uniform’ prior for each of the parameters. For lambda, this is a uniform distribution between 0 and 1, meaning we think all potential values are equally likely. To obtain identical results we will set ‘random.start=FALSE’, if this is set to TRUE a random start value is taken from the system time

```
set.seed(20) # set seed so run will be identical - for example use only
lambda.mcmc <- transformPhylo.MCMC(y=male.length.clade, phy=phy.clade, model="lambda", mcmc.iteration=1000, random.start=FALSE)
```

We can now check the posterior estimate of lambda and convergence of the model. The median and 95 Highest Posterior Density (HPD) is output by the model. Some diagnostics are output as standard: Effective Sample Size (ESS) and acceptance rate. We aim for an ESS of at least 200 and an acceptance rate around 0.44

```
lambda.mcmc[1:4]
```

```
## $median
##      Lambda
## 0.7719967
##
## $`95.HPD`
## lower 95% HPD upper 95% HPD
##      0.5418168      0.9636153
##
## $ESS
##      Lambda
## 218.071
##
## $acceptance.rate
## [1] 0.4306326
```

Our lambda median value is 0.77 but there is a large 95% HPD (0.54-0.96). The ESS and acceptance rate look ok. We can also plot the trace from the MCMC chain - this could look better - running for more generations would help

```
plot(lambda.mcmc$mcmc.chain, type="l", ylim=c(0, 1), xlab="generations", ylab="lambda", las=1)
```

Character displacement models

Magnus Clarke *et al.* (2017) introduced a character displacement model in which inter-specific competition can drive trait change. This model estimates a parameter ‘a’ that drives the strength of inter-specific

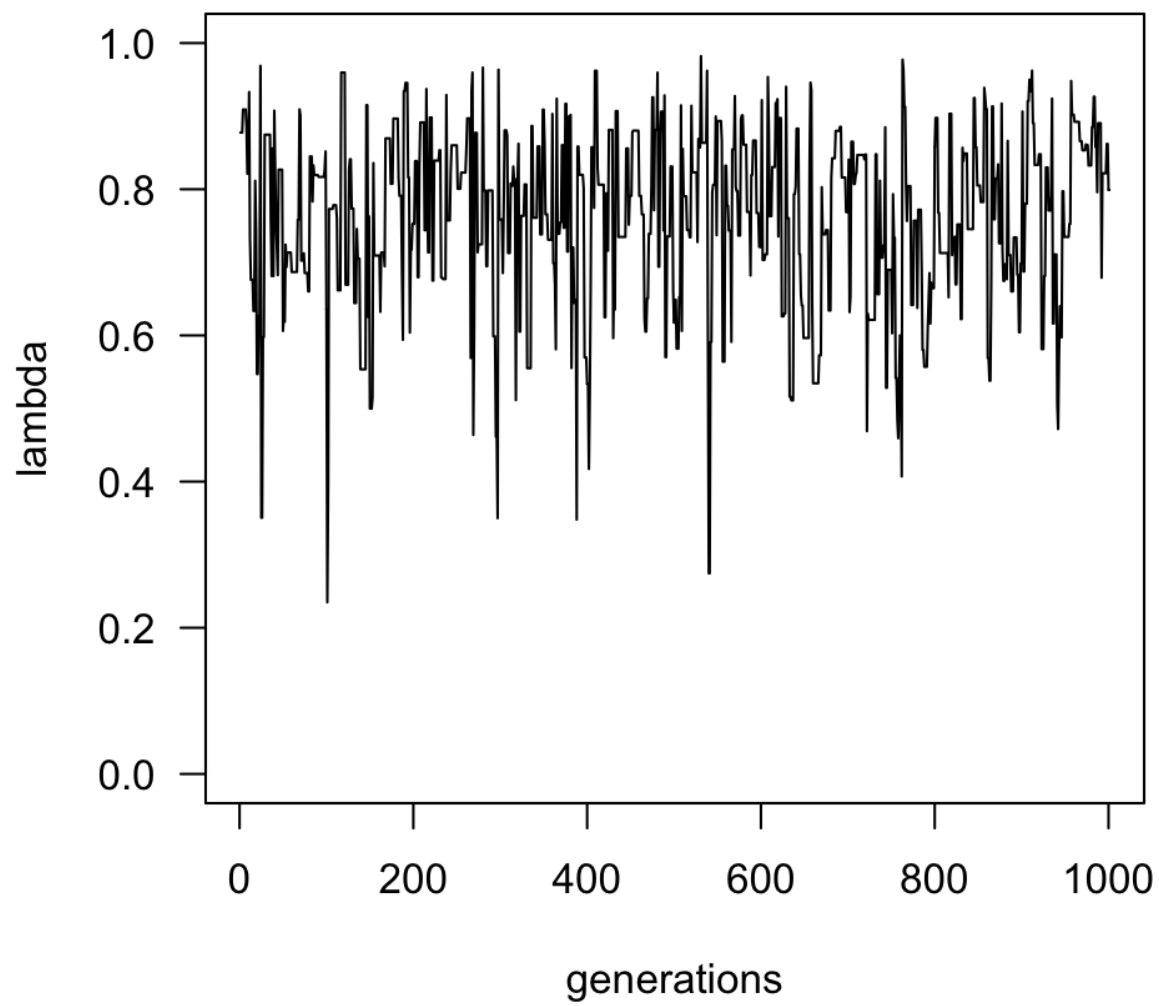


Figure 14: Figure 14. MCMC trace for Pagel's lambda

competition, alongside a Brownian motion model with parameter estimation of the trait variance. If $a=0$ the model is equivalent to Brownian motion, and larger values of a drive trait evolution away from the values of inter-specific competitors.

The character displacement model employs an approximate Bayesian computation (ABC) approach, in which many datasets are simulated based on the known tree using a range of parameter values for ‘ a ’ and the trait variance. These simulations then are compared to the empirical data to estimate the ‘best-fitting’ parameters of the Brownian motion process variance, and the character displacement parameter ‘ a ’.

First data are simulated on the known tree, allowing for a range of variance (sigma) and ‘ a ’ values with both sample from a uniform distribution between 0 and 8. For brevity, we will use 10 simulations only. For actual analyses, many more iterations would be required, perhaps 1 million (Clarke *et al* 2017). Note this process can be made parallel on Mac and Linux systems by using the ‘mc.cores’ argument, but here we will use one core only.

```
data(finches)
emp.tree <- finch.tree
emp.data <- finch.data
param.simulation <- chr.disp.param(emp.tree, n.sim = 10, max.sigma = 8, max.a = 8, ntraits=1, mc.cores = 1)
```

We can then compare these simulated data with the empirical data using the function ‘chr.disp.lrt’. We will use only 5 simulations from the posterior, this value can be guided by simulations (see Clarke *et al.* 2017)

```
chr.disp.lrt(emp.tree=emp.tree, emp.data=emp.data, param.out=param.simulation, posteriorSize=5)
```

```
## $estimates
##           h.0.est  h.1.est
## sigma 2.186667 1.973333
## a      0.000000 2.880000
##
## $likelihood
##           h.0.lik  h.1.lik likelihood.ratio.test
## 1 0.0436706 0.06160842                0.6882469
```

The output shows the ‘estimates’ for hypothesis 0 (Brownian motion) and hypothesis 1 (character displacement) with the variance and a values summarised (a is 0 in the Brownian motion model, by definition). The second list element ‘likelihood.ratio.test’ shows the likelihood of each model, and the value of the likelihood-ratio test statistic.

Fast estimation of Phylogenetic Generalised Least Squares

The package *caper* (Orme *et al* 2018) offers an excellent model to run Phylogenetic Generalised Least Squares (PGLS) models, but these are based-upon Generalised Least Squares (using variance-covariance matrices) which are substantially slower than using independent contrasts (Freckleton 2012).

In *motmot*.2.0, code allows for continuous PGLS models can be estimated using contrasts - this gives identical results to *caper* but is substantially faster, as is shown below. At current only continuous data is allowed in the models for *motmot*.2.0, so if any of the input data are not continuous CAPER or similar should be used. Additionally *motmot*.2.0 only estimates Pagel’s lambda rather than other models, such as Kappa as offered by CAPER

```
# Data and phylogeny
data(anolis.tree)
anolis.tree$node.label <- NULL
lm.data <- transformPhylo.sim(phy=anolis.tree, n=2, model="bm")
dat <- data.frame(x = lm.data[,1], y = lm.data[,2], names = anolis.tree$tip, row.names = anolis.tree$tip)
```

```
comp.dat <- comparative.data(anolis.tree, dat, names)
# pgls from CAPER with matrix inversion
time.now <- Sys.time()
matrix.inv.caper <- pgls( y ~ x, data = comp.dat, lambda="ML")
pgls.time <- Sys.time() - time.now
pgls.time
```

Time difference of 0.6593938 secs

```
time.now <- Sys.time()
picModel <- pic.pglsl(formula=y ~ x, phy=analis.tree, y = dat, lambda="ML", return.intercept.stat=FALSE)
pic.time <- Sys.time() - time.now
pic.time
```

Time difference of 0.128036 secs

The results are identical between the two methods

```
# from caper
summary(matrix.inv.caper)
```

```
##
## Call:
## pgls(formula = y ~ x, data = comp.dat, lambda = "ML")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.57170 -0.72042 -0.04419  0.63206  2.49258
##
## Branch length transformations:
##
## kappa  [Fix]   : 1.000
## lambda [ ML]   : 1.000
## lower bound : 0.000, p = < 2.22e-16
## upper bound : 1.000, p = 1
## 95.0% CI    : (0.974, NA)
## delta  [Fix]   : 1.000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.188911   2.859866 -0.4157   0.6782
## x            0.091117   0.078780  1.1566   0.2491
##
## Residual standard error: 1.01 on 163 degrees of freedom
## Multiple R-squared: 0.00814, Adjusted R-squared: 0.002055
## F-statistic: 1.338 on 1 and 163 DF, p-value: 0.2491
```

```
# from motmot.2.0
picModel
```

```
## $model
##
## Call:
## lm(formula = formula, data = pic.data, x = TRUE, y = TRUE)
##
## Coefficients:
```

```
##          x
## 0.09112
##
##
## $model.summary
##
## Call:
## lm(formula = formula, data = pic.data, x = TRUE, y = TRUE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3464 -0.6466  0.0643  0.8135  3.2631
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## x  0.09112     0.07878   1.157   0.249
##
## Residual standard error: 1.01 on 163 degrees of freedom
## Multiple R-squared:  0.00814,    Adjusted R-squared:  0.002055
## F-statistic: 1.338 on 1 and 163 DF,  p-value: 0.2491
##
##
## $intercept
## [1] -1.188911
##
## $lambda
## [1] 1
##
## $logLikelihood
##           [,1]
## [1,] -534.215
##
## $AIC
##           [,1]
## [1,] 1069.43
```

References

- Blomberg SP, Garland T, and Ives AR. 2003. Testing for phylogenetic signal in comparative data: behavioral traits more labile. *Evolution* 57, 717–45. (doi:10.1111/j.0014-3820.2003.tb00285.x.)
- Butler MA, and King AA. 2004. Phylogenetic comparative analysis: a modeling approach for adaptive evolution. *The American Naturalist* 164, 683-695.
- Cavalli-Sforza, LL, and Edwards AWF. 1967 Phylogenetic analysis: models and estimation procedures. *Evolution* 21, 550-570.
- Clarke M, Thomas GH, and Freckleton RP. 2017. Trait evolution in adaptive radiations: modeling and measuring interspecific competition on phylogenies. *The American Naturalist* 189, 121-137. (doi:10.1086/689819)
- Cooper N, Thomas GH, Venditti C, Meade A, & Freckleton RP. 2016. A cautionary note on the use of Ornstein Uhlenbeck models in macroevolutionary studies. *Biological Journal of the Linnean Society* 118, 64-77. (doi:10.1111/bij.12701)
- Felsenstein J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. *American journal of human genetics* 25, 471.
- Felsenstein J. 1985. Phylogenies and the comparative method. *The American Naturalist* 125, 1-15.
- Freckleton RP. 2012. Fast likelihood calculations for comparative analyses. *Methods in Ecology and Evolution* 3, 940-947. (doi:10.1111/j.2041-210X.2012.00220.x)

- Hansen TF, 1997. Stabilizing selection and the comparative analysis of adaptation. *Evolution* 51, 1341-1351.
- Harmon LJ, *et al.* 2010. Early bursts of body size and shape evolution are rare in comparative data. *Evolution* 64, 2385–96. (doi:10.1111/j.1558-5646.2010.01025.x.)
- Ingram T. 2011. Speciation along a depth gradient in a marine adaptive radiation. *Proceedings of the Royal Society of London B: Biological Sciences* 278, 613-618. (doi:10.1098/rspb.2010.1127)
- Ingram T *et al.* 2016. Comparative tests of the role of dewlap size in Anolis lizard speciation. *Proceedings of the Royal Society of London B: Biological Sciences*, 283, 20162199. (doi:10.1098/rspb.2016.2199)
- O’Meara BC, Ané C, Sanderson MJ, and Wainwright PC. 2006. Testing for different rates of continuous trait evolution using likelihood. *Evolution* 60, 922–33. (doi:10.1111/j.0014-3820.2006.tb01171.x.)
- Orme D, Freckleton RP, Thomas GH, Petzoldt T, Fritz S, Isaac N, and Pearse W. 2018. caper: Comparative Analyses of Phylogenetics and Evolution in R. R package version 1.0.1. <https://CRAN.R-project.org/package=caper>
- Paradis E, Schliep K, and Schwartz R. 2018. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics*. (doi:10.1093/bioinformatics/bty633)
- Pagel, M. Inferring evolutionary processes from phylogenies. 1997. *Zoologica Scripta* 26, 331-348.
- Pagel, M. 1999. Inferring the historical patterns of biological evolution. *Nature* 401, 877.
- Puttick, MN. 2018. Mixed evidence for early bursts of morphological evolution in extant clades. *Journal of Evolutionary Biology* 31, 502-515. (doi:10.1111/jeb.13236.)
- Thomas GH, and Freckleton RP. 2012. MOTMOT: Models of trait macroevolution on trees. *Methods in Ecology and Evolution* 3, 145–51. (doi:10.1111/j.2041-210X.2011.00132.x.)
- Thomas GH, Freckleton RP, and Székely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B: Biological Sciences* 273, 1619–24. (doi:10.1098/rspb.2006.3488.)