

# MOTMOT: Models Of Trait Macroevolution On Trees

Mark Puttick

20 March 2019

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-Deacceleration (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 testing multiple shift times or those selected by the user
- modeSlice fit models in which modes change at a specific time(s) in an extension to models introduced by Slater (2013)
- Nested Shift modes Fit models in which the ancestral BM rate switches to a 'nested' rate within a monophyletic clade in the phylogeny (Puttick 2018)
- 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 install

```
## check for devtools and install if necessary
if (!any(rownames(installed.packages()) == "devtools")) install.packages("devtools")
library(devtools)
install_github("PuttickMacroevolution/motmot", quiet = TRUE)
```

and load MOTMOT

```
library(motmot)
```

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)
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`. Here, we construct a matrix of just `Male_SVL` data, remove missing data, and log-transform the values. All this can be done using the function `sortTraitData`

```
sortedData <- sortTraitData(phy = anolis.tree, y = anolis.data,  
  data.name = "Male_SVL", pass.ultrametric = TRUE)  
phy <- sortedData$phy  
male.length <- sortedData$trait
```

Finally, we will ‘prune’ the species from the tree using `drop.tip` from APE. We plot our tree and data using the `MOTMOT traitData.plot` function.

```
traitData.plot(y = male.length, phy, lwd.traits = 2, col.label = "#00008050",  
  tck = -0.01, mgp = c(0, 0.2, 0), cex.axis = 0.5, show.tips = FALSE)
```

We fit the `tm2` model that allows for clade- and branch-specific changes in rate. Again using the function `transformPhylo.ML`. We fit the models to a subset of these data: including the clade from node 182 only using the APE 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 modelled under the assumption of 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 lineages split on a tree, trait evolution is shared but following a split individual branches evolve and accumulate trait variance independently from their shared ancestral value.

The function `transformPhylo.ML` is used to fit Brownian motion models and its derivatives. Here we fit a simple Brownian motion model to the subset of anolis male SVL data to obtain the Brownian variance, ancestral estimate, log-likelihood, Akaike Information Criterion (AIC), and small-sample AIC (AICc).

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

```
## $brownianVariance  
##           [,1]  
## [1,] 0.001858067  
##  
## $logLikelihood  
## [1] -0.2838382
```

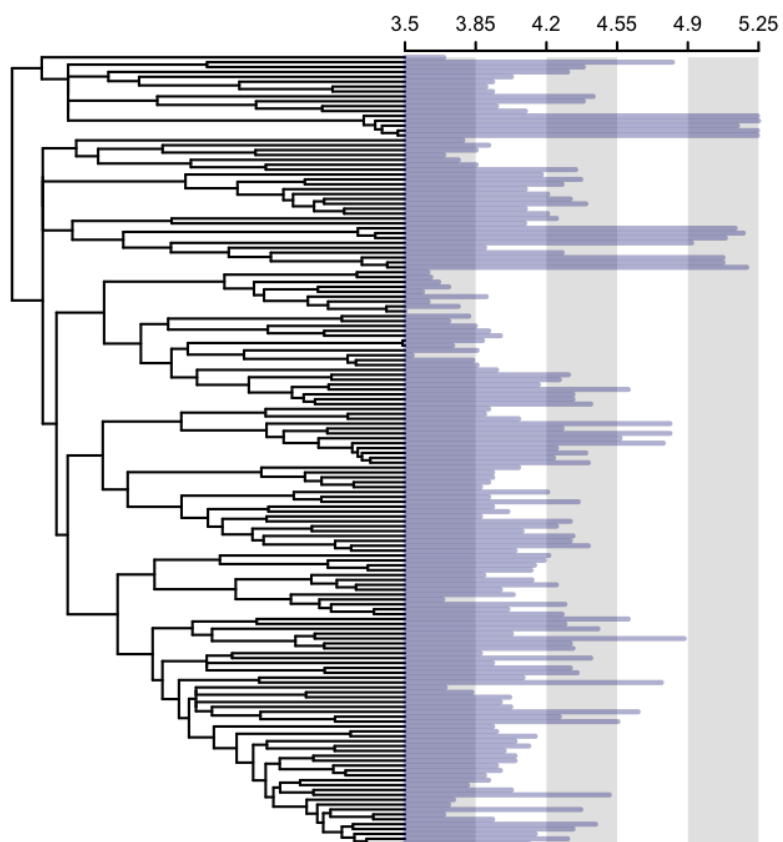


Figure 1: TraitData showing the relative male snout ventral length at the tips

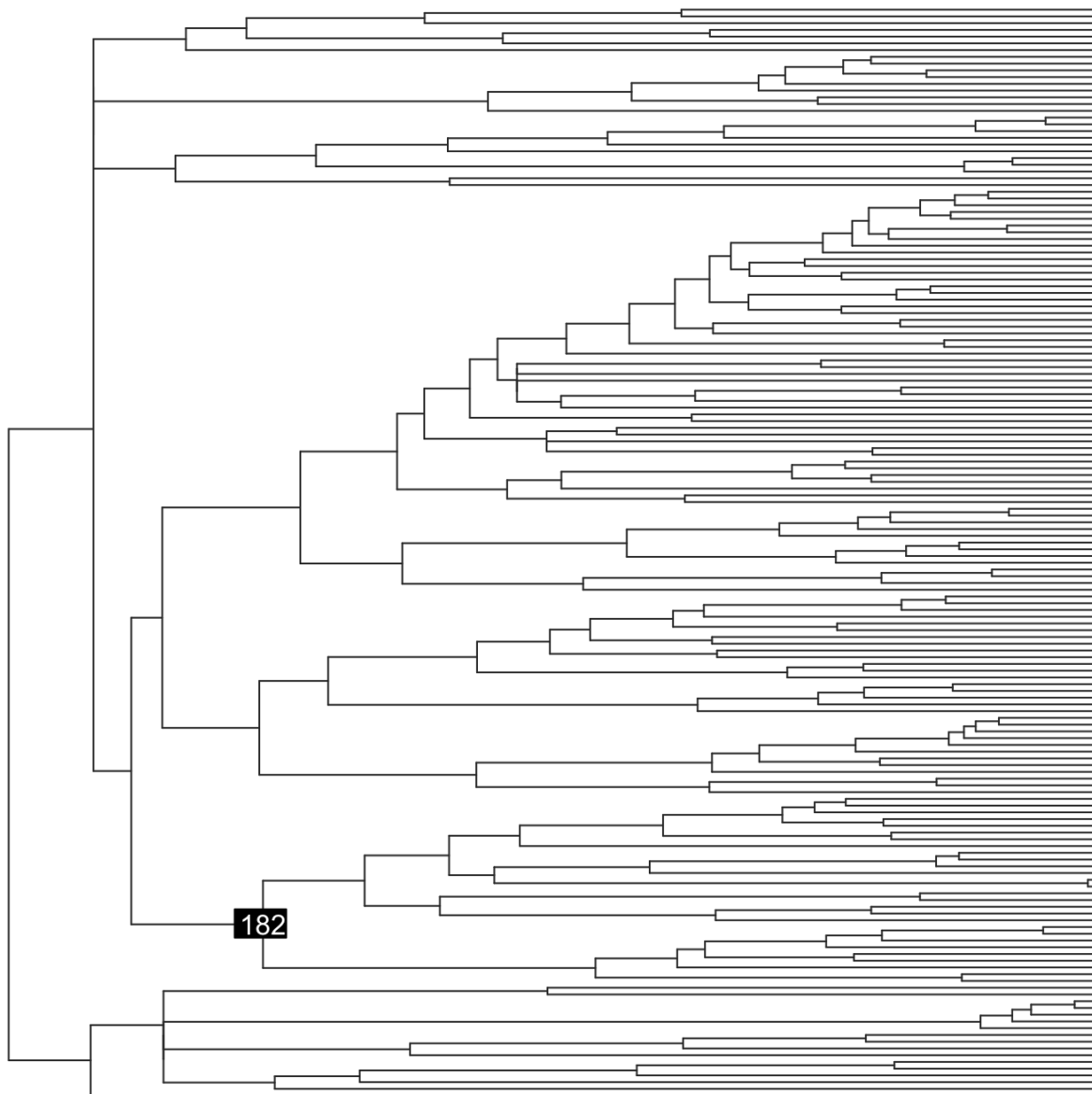


Figure 2: Subset of the anolis tree

```
##
## $root.state
## [1] 3.849481
##
## $AIC
## [1] 4.567676
##
## $AICc
## [1] 5.047676
```

## Pagel's lambda

Here we 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 transformed by the Lambda parameter value. When the parameter Lambda equals 1, branches are transformed by multiplying by 1 and so the model is equal to Brownian motion (high phylogenetic signal). Values of Lambda under 1 suggest there has been less influence of shared history on trait values at the tips. Finally, a Lambda 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

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

```
## $MaximumLikelihood
## [1] 6.522191
##
## $Lambda
##      MLLambda   LowerCI   UpperCI
## [1,] 0.8369974 0.5259423 0.9742338
##
## $brownianVariance
##              [,1]
## [1,] 0.0008245359
##
## $root.state
## [1] 3.853432
##
## $AIC
## [1] -7.044383
##
## $AICc
## [1] -6.044383
```

The maximum likelihood estimate of Pagel's Lambda is equal to 0.84.

A new feature in MOTMOT allows for plotting of the likelihood profile for the branch-transformation parameter, in this case Pagel's lambda using the argument `profilePlot` in `transformPhylo.ML`.

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

We now compare the relative fit of the BM and Lambda models. Lambda has higher likelihood, but it also has more parameters. The root state and sigma-squared (rate) parameters are present in both models but the

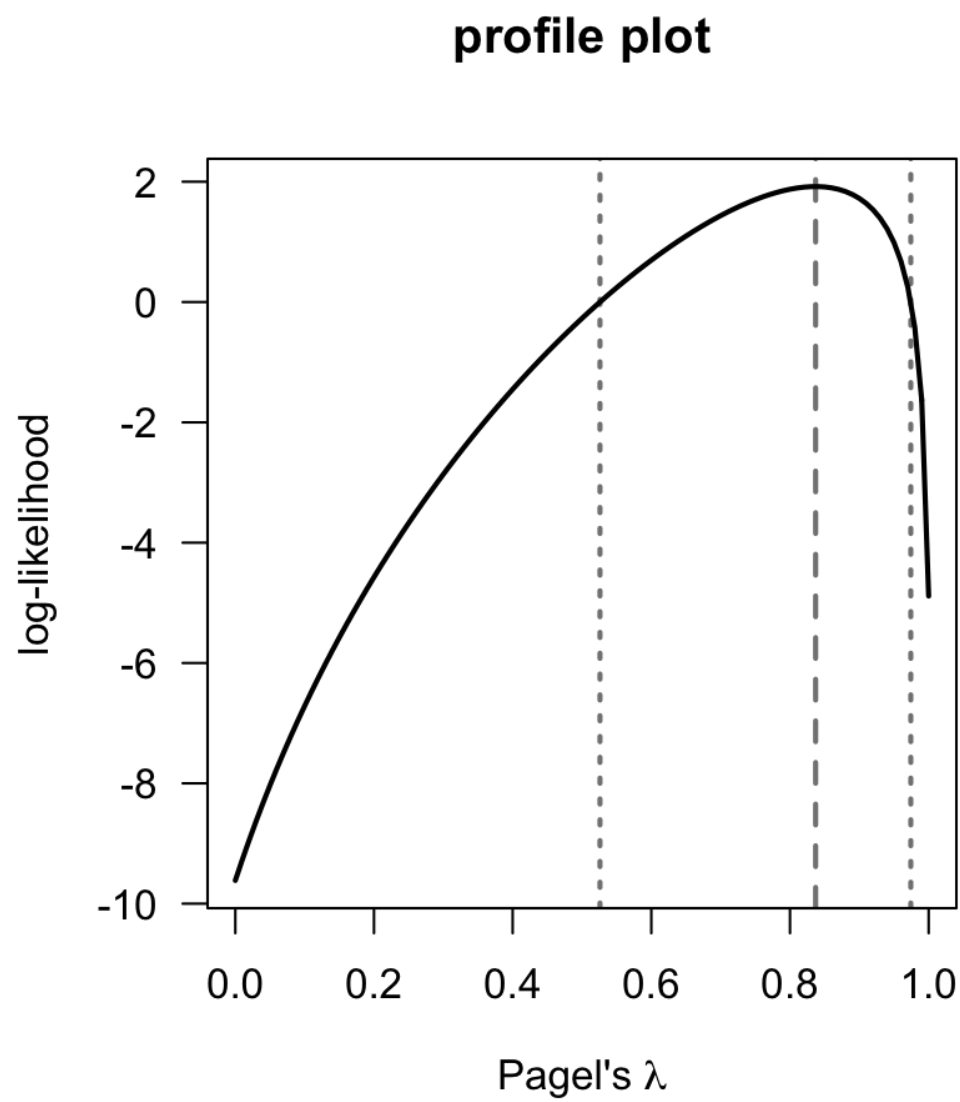


Figure 3: Profile plot of ML estimation for Pagel's lambda

Lambda model also requires an estimate of the parameter Lambda. We can test whether the Lambda model is a significant improvement over BM. First we test the relative fit by using the chi-squared distribution. The models differ in one degree of freedom: BM has 2 parameters and Lambda has 3. We can use the `stats` function `pchisq` to obtain a p value by testing using a chi-squared distribution. The Lambda is indeed a superior fit compared to BM when fit to these data ( $p < 0.05$ ).

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

```
## [1] 0.009085056
```

Additionally there is a large small-sample 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] 11.09206
```

## 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 Maximum likelihood estimated for Delta of 2.23 but the CI spans  $< 1$  to  $> 4$ , so it is not possible to conclusively support a change in the rate of evolution through time.

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

```
delta.ml
```

```
## $MaximumLikelihood
## [1] 1.179797
##
## $Delta
##      MLDelta   LowerCI   UpperCI
## [1,] 2.231464 0.8477531 4.660712
##
## $brownianVariance
##           [,1]
## [1,] 6.013992e-06
##
## $root.state
## [1] 3.8843
##
## $AIC
## [1] 3.640407
##
## $AICc
## [1] 4.640407
```

## Kappa

Kappa is used as a measure of punctuated evolution and spans values of 0-1 (Pagel 1997:1999). A Kappa value of 1 is equivalent to BM, and 0 indicates trait change occurs at events of speciation. Here there is

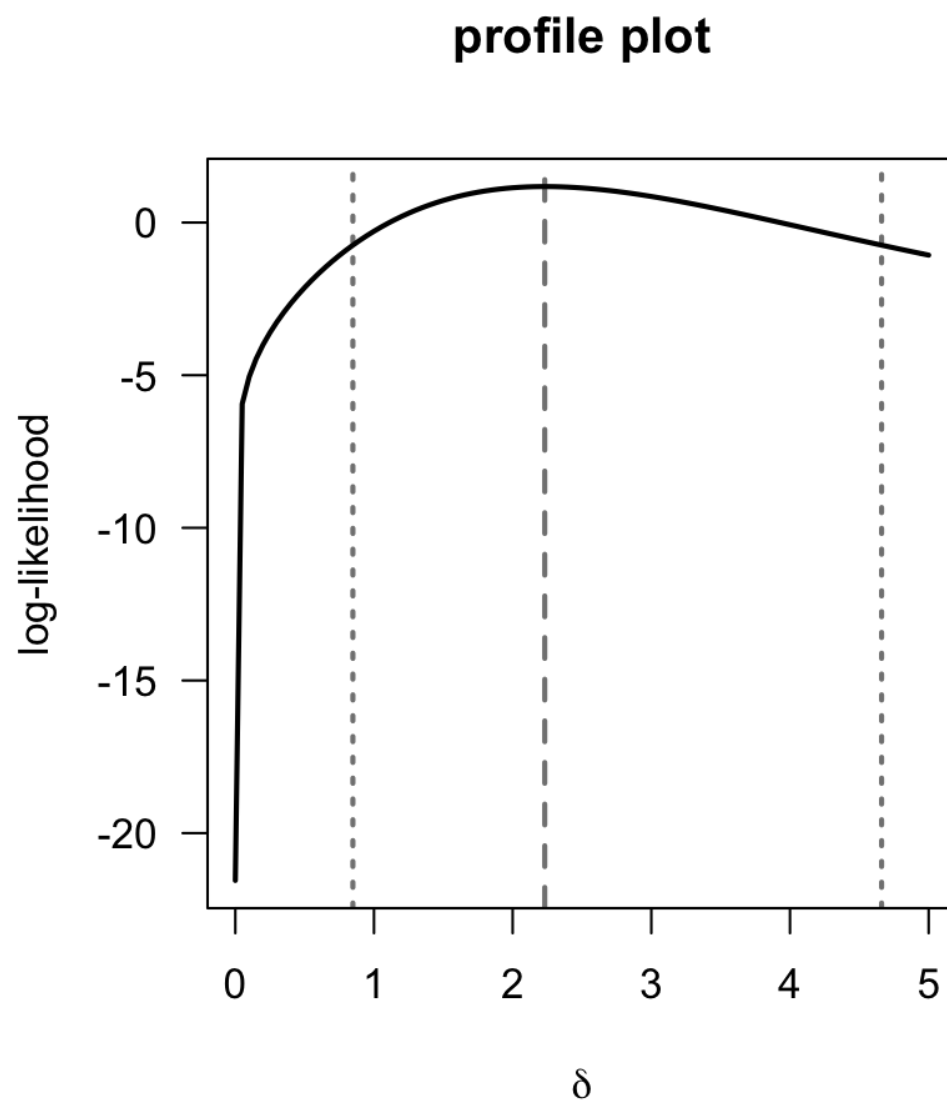


Figure 4: Profile plot to estimate delta



evidence of punctuated evolution. `transformPhylo.ML` also allows users to see the the phylogeny transformed by model parameters. As an example, we show the original, BM model phylogeny and compare this with the phylogeny transformed by the Kappa phylogeny.

```
kappa.ml <- transformPhylo.ML(phy = phy.clade, y = male.length.clade,
  model = "kappa", profilePlot = FALSE)
par(mfrow = c(1, 2))
plot.phylo(phy.clade, show.tip.label = FALSE, no.margin = TRUE)
mtext("Original phylogeny", 3, cex = 0.8)
plot.phylo(kappa.ml$kappaPhy, show.tip.label = FALSE, no.margin = TRUE)

## Warning in plot.phylo(kappa.ml$kappaPhy, show.tip.label = FALSE, no.margin
## = TRUE): found less than 2 tips in the tree

## NULL

mtext("Kappa model phylogeny", 3, cex = 0.8)
mtext("Kappa = 1e-8", 3, cex = 0.8, line = -1)
```

## 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, 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] 1.743459
##
## $Alpha
##      MLAlpha      LowerCI      UpperCI
## [1,] 0.01693505 0.0004499009 0.03912295
##
## $brownianVariance
##      [,1]
## [1,] 0.002824059
##
## $root.state
## [1] 3.876704
##
## $AIC
## [1] 2.513082
##
## $AICc
## [1] 3.513082
```

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 so it does not have higher relative support compared to BM ( $p > 0.05$ ).

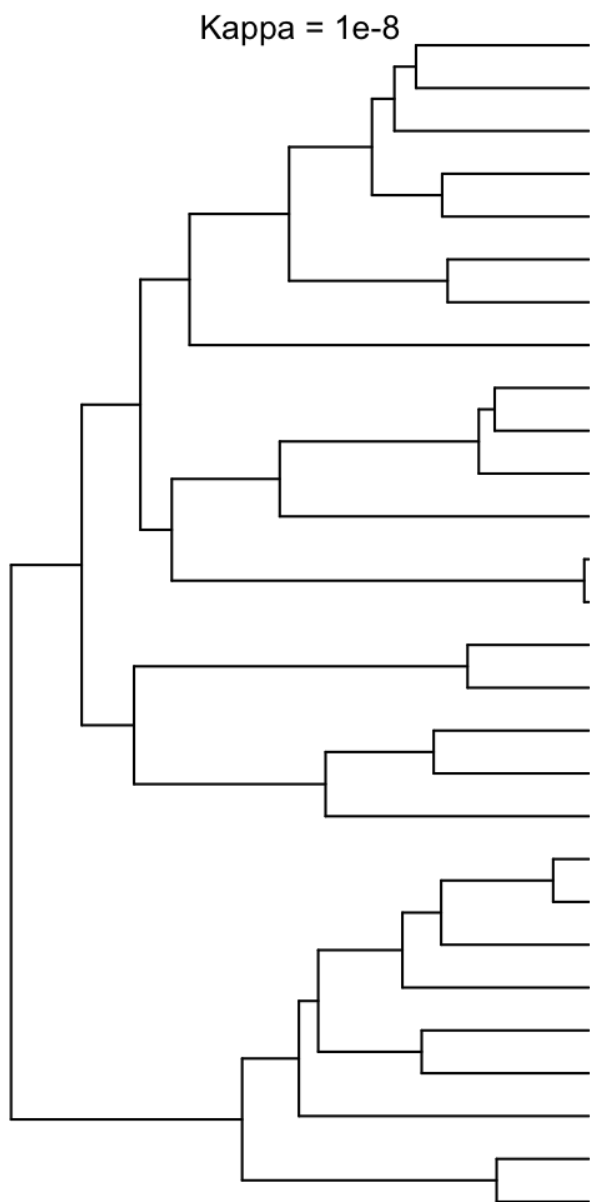


Figure 5: Comparison of BM and Kappa transformed trees.

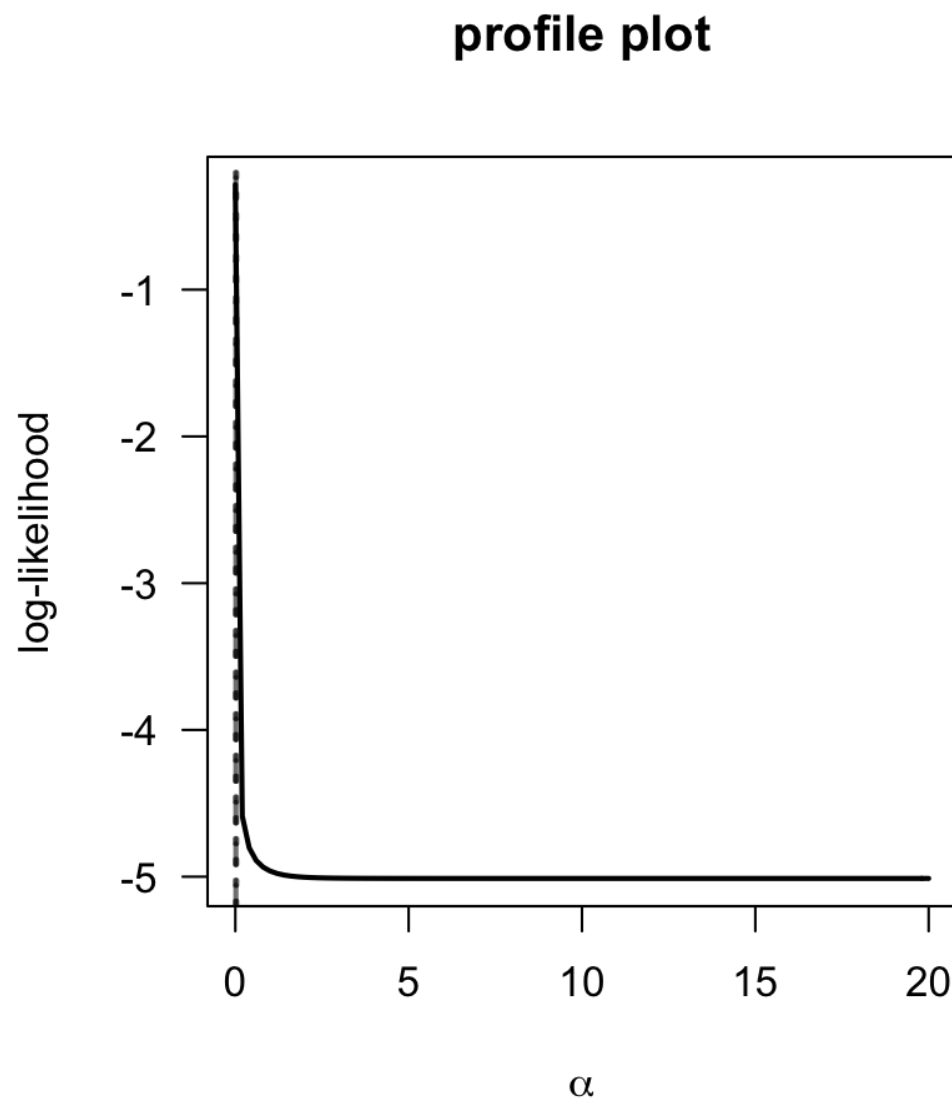


Figure 6: Profile plot to estimate alpha

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

```
## [1] 0.1544952
```

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

```
## [1] 1.534594
```

## 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 = TRUE)
```

```
acdc.ml
```

```
## $MaximumLikelihood
## [1] 1.743459
##
## $ACDC
##           MLacdc      LowerCI      UpperCI
## [1,] 0.03385981 0.000879245 0.04246516
##
## $brownianVariance
##           [,1]
## [1,] 0.0002590746
##
## $root.state
## [1] 3.876696
##
## $AIC
## [1] 2.513082
##
## $AICc
## [1] 3.513082
```

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] 0.1544951
```

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-06, print.warning = FALSE)
```

```
## $MaximumLikelihood
## [1] -0.2839606
```

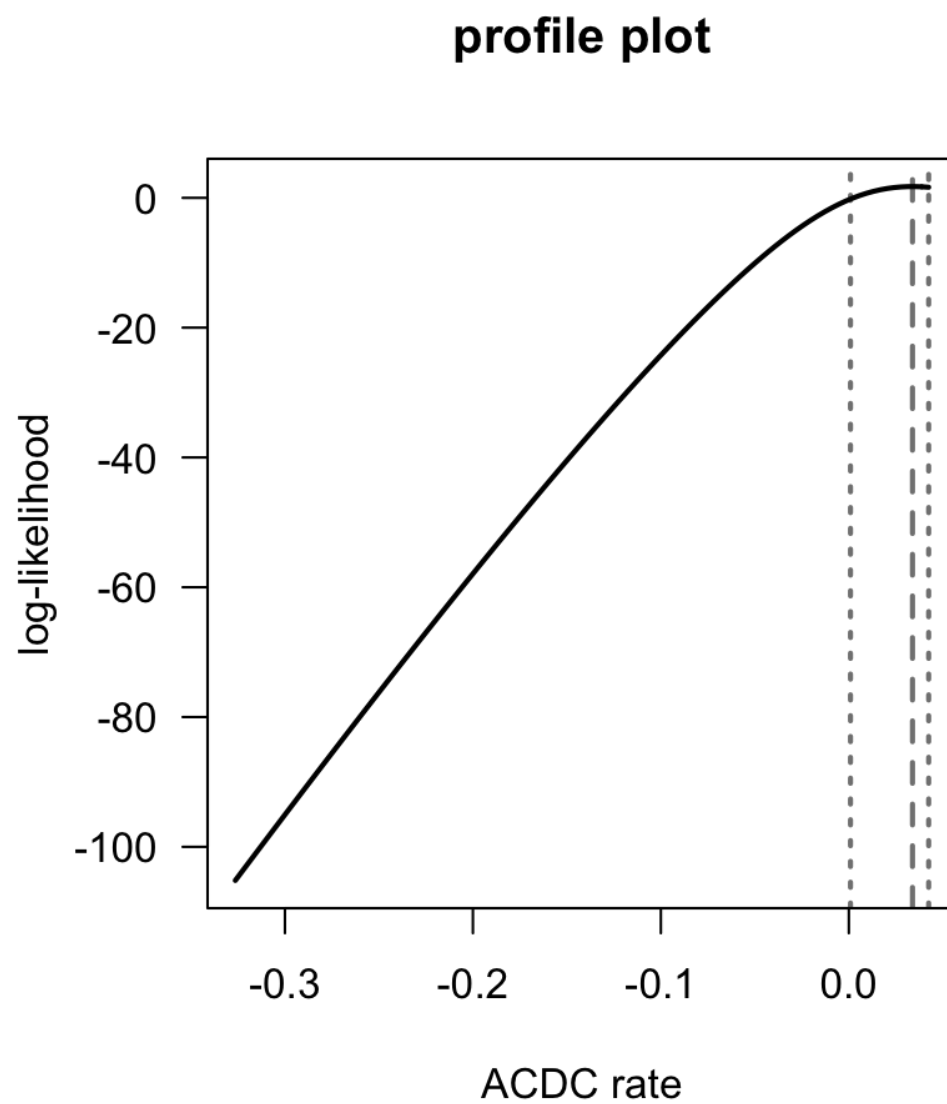


Figure 7: Profile plot to estimate the ACDC parameter

```
##
## $ACDC
##      MLacdc      LowerCI UpperCI
## [1,] -1e-06 -0.01322547 -1e-06
##
## $brownianVariance
##      [,1]
## [1,] 0.001858181
##
## $root.state
## [1] 3.849481
##
## $AIC
## [1] 6.567921
##
## $AICc
## [1] 7.567921
```

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

## psi and multispi

The parameter psi is similar to the parameter Kappa in that it is a measure of the relative contribution of speciational (~punctuated) and gradual evolution to trait change on a phylogeny (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)
```

```
psi.ml
```

```
## $MaximumLikelihood
## [1] 8.495569
##
## $psi
##      MLpsi LowerCI UpperCI
## [1,]      1 0.316058      1
##
## $brownianVariance
##      [,1]
## [1,] 0.0008018518
##
## $root.state
## [1] 3.761269
##
## $AIC
## [1] -10.99114
##
## $AICc
```

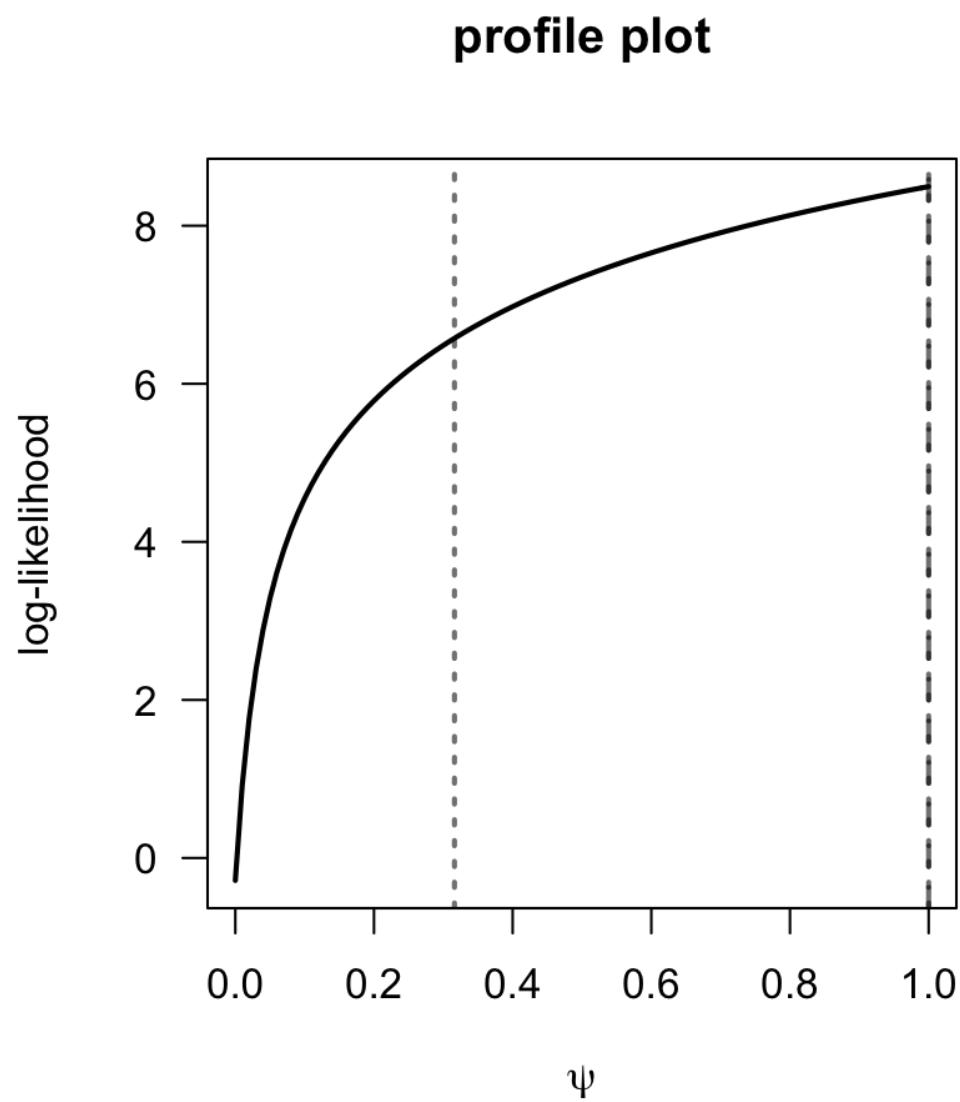


Figure 8: Profile plot to estimate the psi parameter

```
## [1] -9.991139
```

This indicates support for the psi model but it is not a significant improvement on Brownian motion ( $p > 0.05$ ).

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

```
## [1] 0.1544951
```

We could also get a potentially more accurate of speciation rates by using the full tree, rather than the pruned tree to estimate speciation and extinction rates as this will give more accurate estimates rather than using the taxa with complete data only. If extinction rates are larger than 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 = TRUE, full.phy = phy)
```

```
## $MaximumLikelihood
## [1] 8.495569
##
## $psi
##      MLpsi  LowerCI UpperCI
## [1,]      1 0.316058      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. We can now fit the multipsi model with these data. In MOTMOT, this model requires branch labels given *a priori* by the user to delimit the different regimes on the phylogeny. Note that these clades with potentially different psi regimes do not need to be monophyletic clades. Here we arbitrarily assign two clades 'a' and 'b' to test differences between them.

```
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")
```

Using these data we fit the model with transformPhylo.ML.

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

```
## $MaximumLikelihood
```



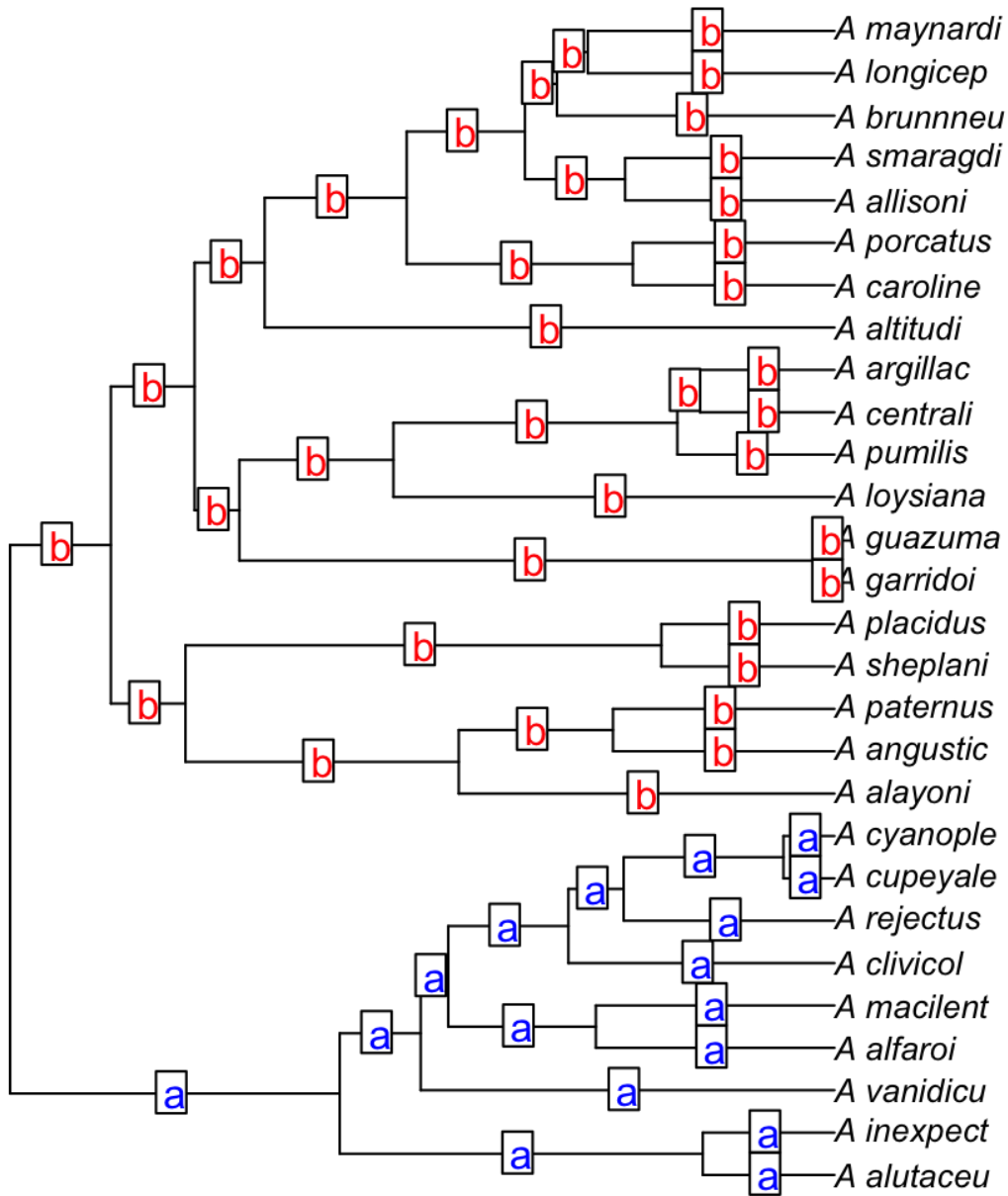


Figure 9: Two clades used in the multipsi model

```
## [1] 8.495569
##
## $psi
##      MLpsi      LowerCI UpperCI
## a      1 0.04812257      1
## b      1 0.20955316      1
##
## $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

## Estimate Pagel's Lambda Alongside Other Modes

One way to deal with 'noisy' data is to estimate Pagel's lambda alongside a parameter of interest. By using Pagel's Lambda alongside other models it may be possible to account for variation in the data that may be due to errors in the phylogeny or the trait data. In MOTMOT, Pagel's 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] 1.743459
##
## $ACDC
##      MLacdc      LowerCI      UpperCI
## [1,] 0.03385981 0.000879245 0.04246516
##
## $brownianVariance
##              [,1]
## [1,] 0.0002590746
##
## $root.state
## [1] 3.876696
##
## $AIC
## [1] 2.513082
##
## $AICc
## [1] 3.513082
```

```
# ACDC model plus lambda
acdc.ml.lambda
```

```
## $MaximumLikelihood
## [1] 7.376867
##
## $ACDC
##           MLacdc    LowerCI    UpperCI
## [1,] -0.1828599 -0.3242442 -0.08285734
##
## $brownianVariance
##           [,1]
## [1,] 0.01271863
##
## $root.state
## [1] 3.83604
##
## $lambda
## [1] 0.1389661
##
## $AIC
## [1] -4.753735
##
## $AICc
## [1] -2.026462
```

We can see lambda is  $< 1$ , and this has affected the parameter estimation. The improvement in the model fit is significant compared to the ACDC model fit without estimating Lambda 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] 0.01762134
```

```
# 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] 0.02170196
```

## 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. First, 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")
```

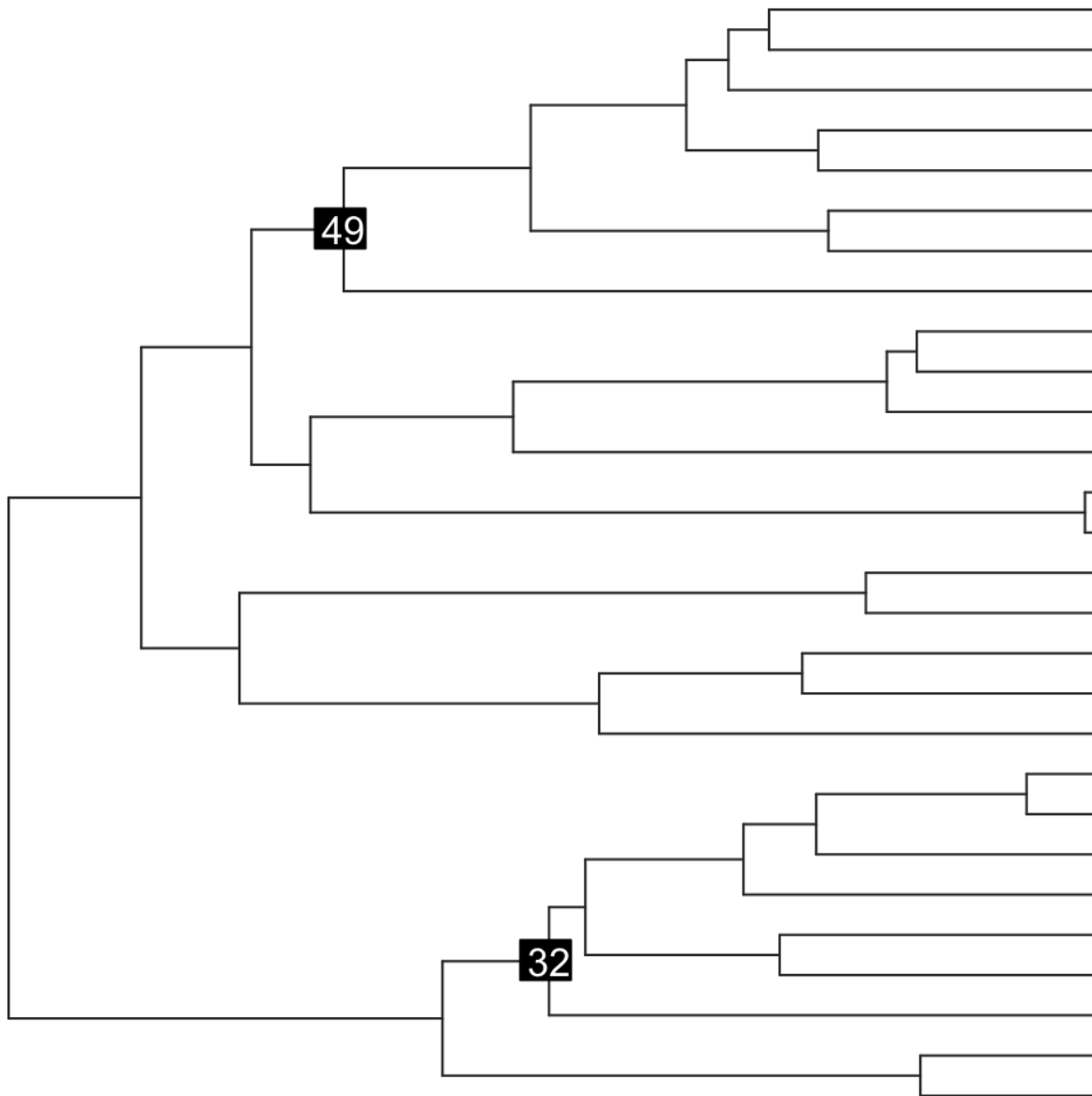


Figure 10: Lineages with different rates of evolution

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 0.8138100 0.2632955 3.263628
## [2,]    49 0.6819079 0.1896347 2.952364
##
## $MaximumLikelihood
## [1] -0.1462557
##
## $brownianVariance
##      [,1]
## [1,] 0.002143258
##
## $root.state
## [1] 3.870488
##
## $AIC
## [1] 8.292511
##
## $AICc
## [1] 10.03164
```

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.

## 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 the `traitMedusa` model in which nodes are individually tested for rate increases or decreases, and the clade and/or branch with a rate change that produces the largest increase in likelihood is returned. Note, 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=3)
# 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 -0.2838382        2 4.567676 5.047676
##
```

```
## Shift 1
##      node shiftPos      lnL n.params      AIC      AICc      rate.1
## shift.1   39      clade 3.042358          3 -0.08471602 0.915284 0.09148646
##
## Shift 2
##      node shiftPos      lnL n.params      AIC      AICc      rate.1
## shift.2   44      clade 4.746785          5 0.5064296 3.233702 0.1408068
##      rate.2
## shift.2 3.158565
```

We can summarise the analyses using `traitMedusaSummary` and plotting the shifts on the phylogeny using the function `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 3.042358          3 -0.08471602 0.915284
##
## $Rates
##      node shiftPos      MLRate      LowerCI      UpperCI
## 1   39      clade 0.0914864604723702 0.0260301808222033 0.500374159059036
##
## $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 = "rates", type = "fan", cex = 0.5, edge.width = 2)
```

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 can all be calculated with the MOTMOT function `calcCutOff`. The function requires the tree and input from the model applied to the empirical data as well as the number of simulations. Here we calculated the AICc cut-off required for the `tm2` analysis from above.

```
# uncomment to run set.seed(203) calcCutOff(phy.clade,
# n=1000, model='tm2', minCladeSize=5, nSplits=1) 95%
# 5.698198
```

Here if we repeat this analysis with the appropriate AICc cut-off (5.698) then we see that the single-rate Brownian motion is, in fact, supported.

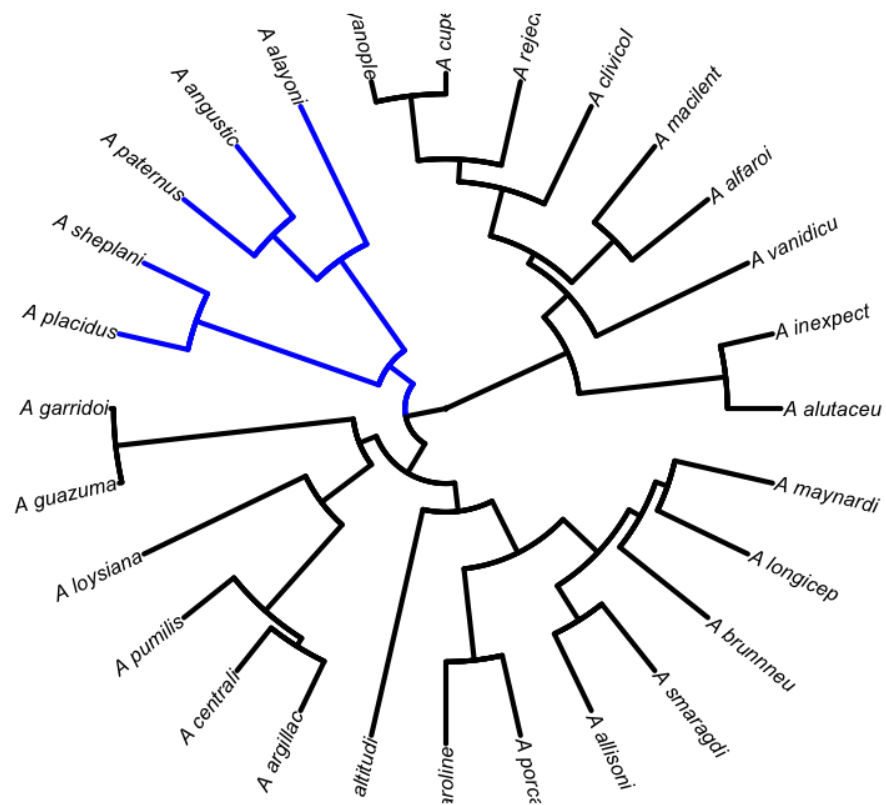


Figure 11: The subset of the tree showing the rate heterogeneity estimated from the traitMedusa model

```
traitMedusaSummary(tm2.ml, cutoff = 5.698198, AICc = TRUE)$Rates
```

```
## [1] "Single rate"
```

## timeSlice 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(10))
```

```
## [1] "BM model"
##          lnL          AIC          AICc    sigma.sq.1  anc.state.1
## -0.283838169  4.567676338  5.047676338  0.001858067  3.849481405
## [1] "shiftModel"
##          lnL          AIC          AICc    sigma.sq.1  anc.state.1
##  2.946497537  2.107004926  3.846135361  0.001006387  3.860015270
##          rates1      rates2  time.split1
##  0.692073080  2.944764076  10.000000000
```

We can use the function `timeSliceSummary` to summarise and plot the results. The output summarises the best model according to AICc fit. 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.ramp = c("blue", "red"))
```

```
## Warning in plot.window(...): "cex.tip" is not a graphical parameter
## Warning in plot.window(...): "phylo.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "cex.tip" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "phylo.width" is not a graphical
## parameter
## Warning in title(...): "cex.tip" is not a graphical parameter
## Warning in title(...): "phylo.width" is not a graphical parameter
## Warning in plot.window(...): "cex.tip" is not a graphical parameter
## Warning in plot.window(...): "phylo.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "cex.tip" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "phylo.width" is not a graphical
## parameter
## Warning in title(...): "cex.tip" is not a graphical parameter
```



```
## Warning in title(...): "phylo.width" is not a graphical parameter
## Warning in plot.window(...): "cex.tip" is not a graphical parameter
## Warning in plot.window(...): "phylo.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "cex.tip" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "phylo.width" is not a graphical
## parameter
## Warning in title(...): "cex.tip" is not a graphical parameter
## Warning in title(...): "phylo.width" is not a graphical parameter
## Warning in plot.window(...): "cex.tip" is not a graphical parameter
## Warning in plot.window(...): "phylo.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "cex.tip" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "phylo.width" is not a graphical
## parameter
## Warning in title(...): "cex.tip" is not a graphical parameter
## Warning in title(...): "phylo.width" is not a graphical parameter
```

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

```
outputSummary$RatesCI
```

```
##          rates      LCI      UCI
## rates1 0.6920731 0.1873339 2.11563
## rates2 2.9447641 0.9633084 10.87739
```

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, boundaryAge = 8)
```

```
## [1] BM model
##          lnL          AIC          AICc    sigma.sq.1  anc.state.1
## -0.283838169  4.567676338  5.047676338  0.001858067  3.849481405
## [1] shift 1
##          lnL          AIC          AICc    sigma.sq.1  anc.state.1      rates1
## 3.584675298  0.830649404  2.569779838  0.001012562  3.859446535  0.666861158
##          rates2 time.split1
## 3.238675325  8.000000000
## [1] shift 2
##          lnL          AIC          AICc    sigma.sq.1  anc.state.1
## 4.054551e+00  1.890899e+00  4.618171e+00  1.066343e-04  3.856792e+00
##          rates1      rates2      rates3  time.split1  time.split2
## 7.019519e+00  1.000000e-08  3.116130e+01  8.000000e+00  1.354564e+01
```

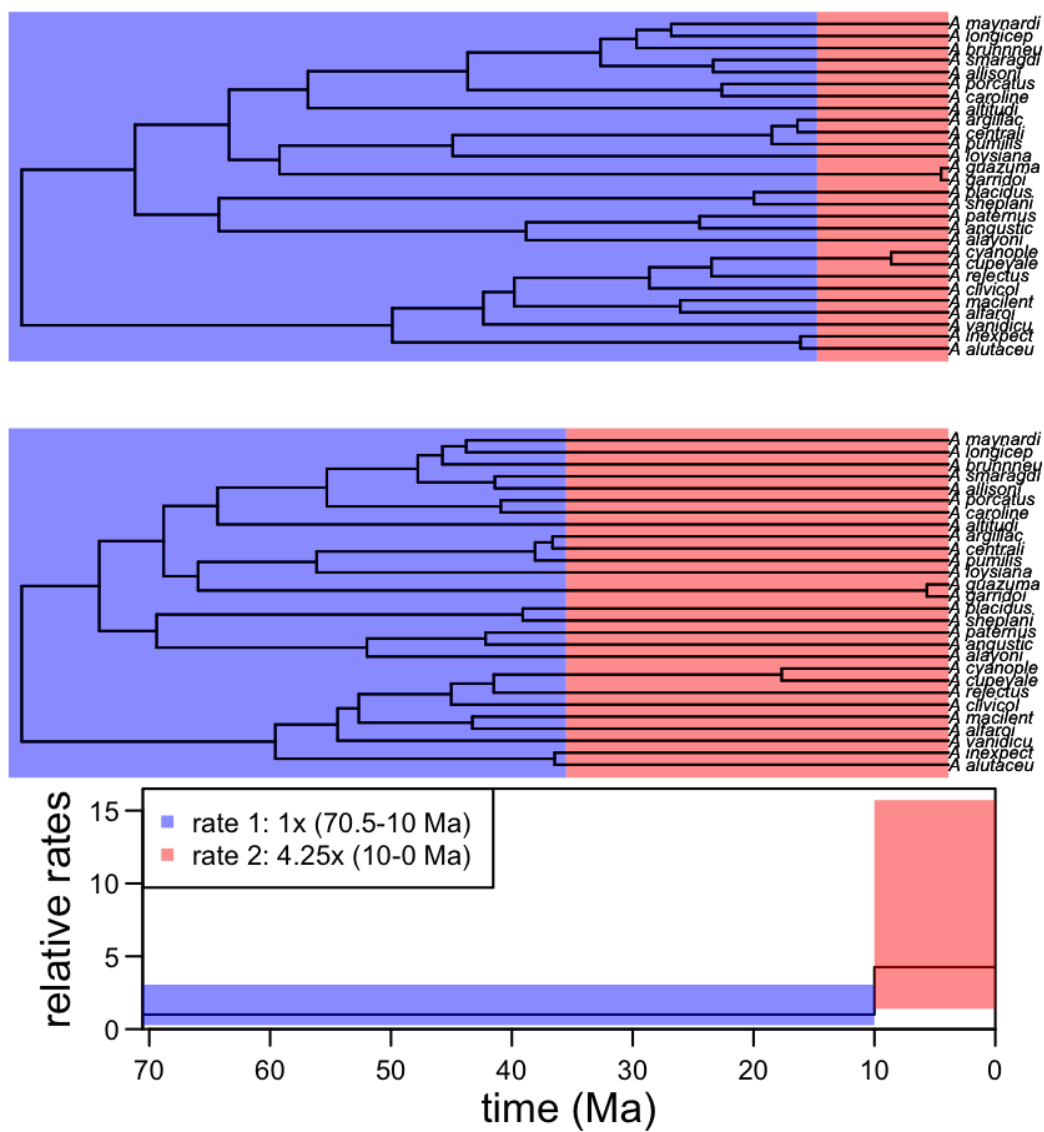


Figure 12: TimeSlice plot with a split at 10 Ma

And summarise the results. We can select 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("blue", "red"))

## Warning in plot.window(...): "cex.tip" is not a graphical parameter
## Warning in plot.window(...): "phylo.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "cex.tip" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "phylo.width" is not a graphical
## parameter
## Warning in title(...): "cex.tip" is not a graphical parameter
## Warning in title(...): "phylo.width" is not a graphical parameter
## Warning in plot.window(...): "cex.tip" is not a graphical parameter
## Warning in plot.window(...): "phylo.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "cex.tip" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "phylo.width" is not a graphical
## parameter
## Warning in title(...): "cex.tip" is not a graphical parameter
## Warning in title(...): "phylo.width" is not a graphical parameter
## Warning in plot.window(...): "cex.tip" is not a graphical parameter
## Warning in plot.window(...): "phylo.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "cex.tip" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "phylo.width" is not a graphical
## parameter
## Warning in title(...): "cex.tip" is not a graphical parameter
## Warning in title(...): "phylo.width" is not a graphical parameter
## Warning in plot.window(...): "cex.tip" is not a graphical parameter
## Warning in plot.window(...): "phylo.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "cex.tip" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "phylo.width" is not a graphical
## parameter
## Warning in title(...): "cex.tip" is not a graphical parameter
## Warning in title(...): "phylo.width" is not a graphical parameter
```

## modeSlice model

In a related extension, we have incorporated the new modeSlice model to the transformPhylo.ML. modeSlice incorporates and extends the methods of Slater (2013) by allowing for multiple shifts in various modes of evolution (BM, OU, EB, and Kappa) at different times in the phylogeny's history. This is flexible as users

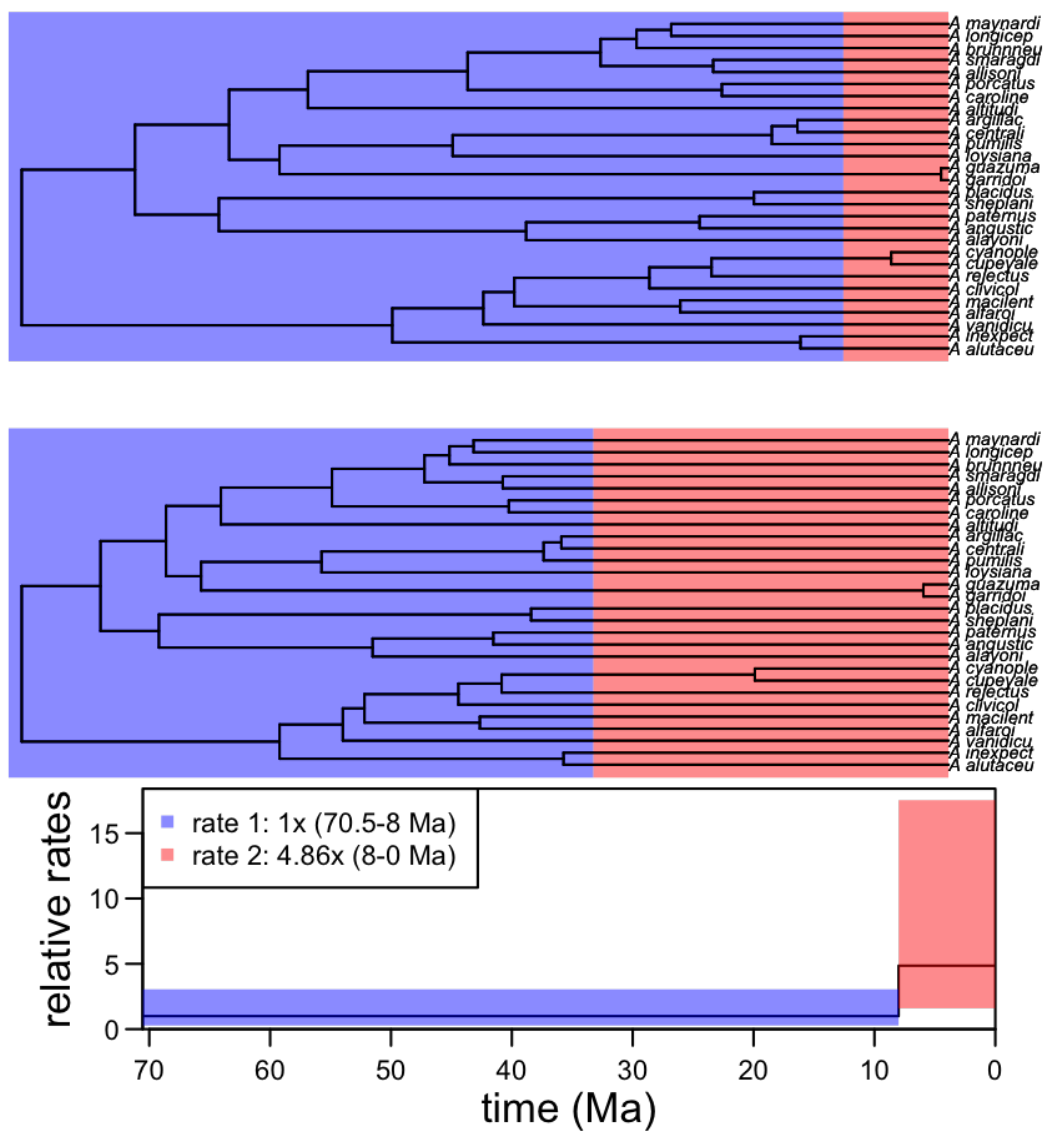


Figure 13: TimeSlice plot with Maximum likelihood estimation of split time

can input multiple rate shift times with different combinations of modes. Furthermore, time bins with a BM mode of evolution can optionally vary in the rate of evolution compared to the background variance (rate.var argument), and users can include a rate scalar alongside EB modes.

Here a model is fit with a shift from an ACDC/EB model with associated rate scalar to an OU model 40 Ma and then to a BM rate shift model at 30 Ma to the present. The results indicate an ACDC/EB scalar (root age-40Ma), followed by a OU model with alpha of 1.75 (40-30Ma), followed by a rate increase (5.3x background from 30-0 Ma). However this model is not supported over Brownian motion.

## Nested models of evolution

We can also test models of nested evolution in which an ancestral model of BM evolution changes to an 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 a significantly better relative fit 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))
nested.ou <- transformPhylo.ML(male.length.clade, phy = phy.clade,
  model = "OU", nodeIDs = c(44))

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

## [1] 0.05740889

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

## [1] 0.3614244
```

## Bayesian estimation of tree transformation models

Parameters of various modes of evolution can be conducted using a simple Bayesian Markov Chain Monte Carlo (MCMC) algorithm in `transformPhylo.MCMC` which may better reflect probabilistic uncertainty in parameter estimates compared to Maximum likelihood estimation. By default the model places a uniform prior `uniform.prior=TRUE` on parameters with a lower bound of zero; these bounds can be left to the default values or customised by the user. Alternatively, users can select a truncated normal prior `uniform.prior=FALSE` as the prior distribution, with user controls on the mean, standard deviation, and lower and upper bounds.

The Bayesian model has an optional built-in optimisation routine to maximise the acceptance ratio prior to the running the MCMC chain `opt.accept.rate`. Users can customise the standard deviation of the proposal distribution and the target acceptance rate. After completion, the function returns convergence diagnostics (effective sample size, acceptance proportion ratio), MCMC chain, and the median value and 95% Highest Posterior Density of the estimated parameter.

The function ‘transformPhylo.MCMC’ allows for the estimation of model parameters using Bayesian statistics. Models of lambda, delta, kappa, OU, ACDC, psi, and multi-psi can currently be modelled using transformPhylo.MCMC. Additionally, Pagel’s Lambda can be optimised alongside parameters and nested modes in the same way as `transformPhylo.ML`.

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, burn.in = 0.1, 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
## 0.7841303
##
## $`95.HPD`
## lower 95% HPD upper 95% HPD
##      0.5285959      0.9718965
##
## $ESS
##      Lambda
## 234.6245
##
## $acceptance.rate
## [1] 0.6304107
```

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 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’.

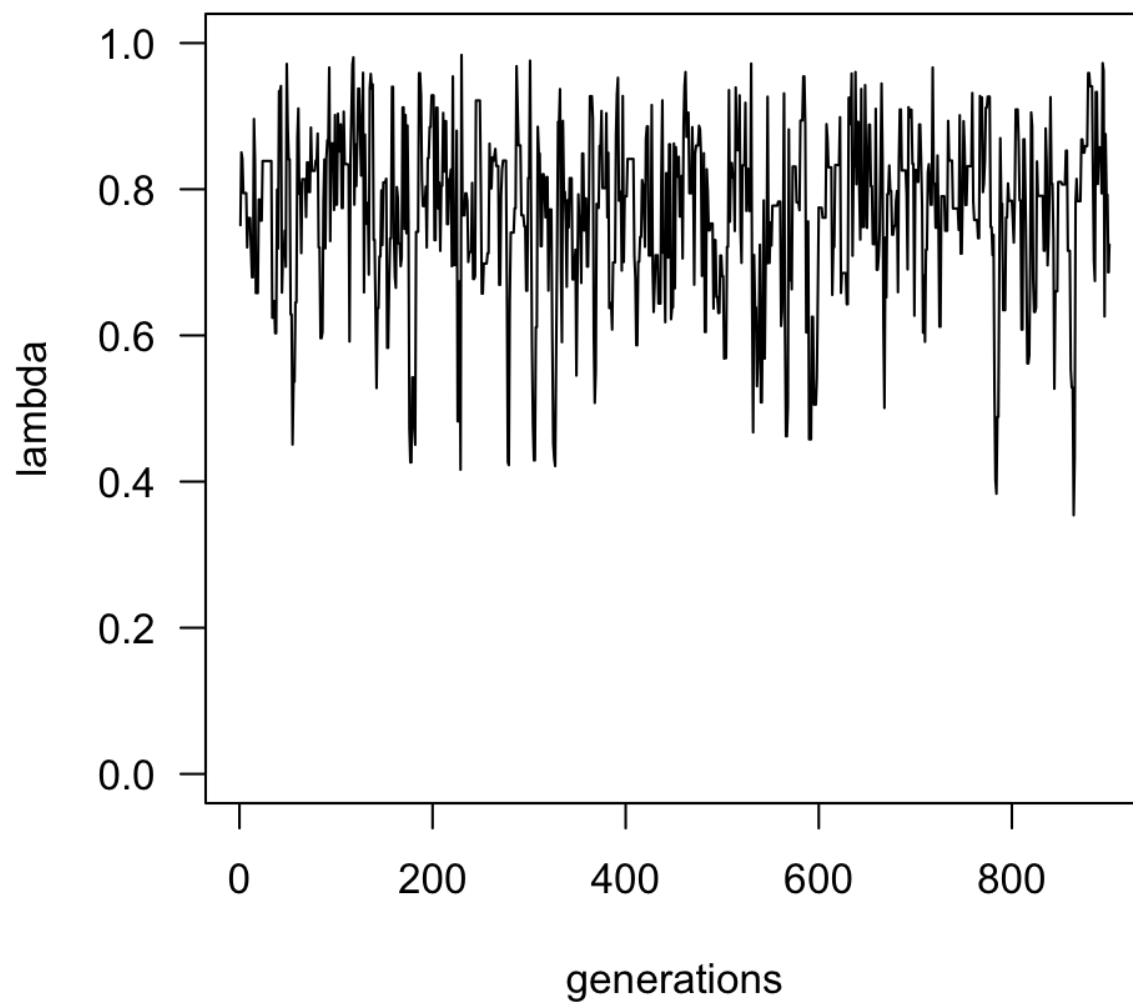


Figure 14: MCMC trace for Pagel's lambda

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 100 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 = 100, 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 75 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 = 75)
```

```
## $estimates
##      h.0.est  h.1.est
## sigma    2.56 2.666667
## a        0.00 5.440000
##
## $likelihood
##  log.h.0.lik. log.h.1.lik. likelihood.ratio.test  p.value
## 1      -4.396987    -3.705833                1.382309 0.7602917
```

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*, 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*, so if any of the input data are not continuous CAPER or similar should be used. Additionally *motmot* only estimates Pagel’s Lambda rather than other models, such as Kappa as offered by CAPER

```
# Data and phylogeny
data(anolis.tree)
anolis.tree$node.label <- NULL
set.seed(3492)
lm.data <- transformPhylo.sim(phy = anolis.tree, n = 2, model = "bm")
dat <- data.frame(x = lm.data[, 1], y = lm.data[, 2], names = anolis.tree$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 0.984056 secs

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

```

## Time difference of 0.2047589 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
## -2.27281 -0.52137  0.08971  0.75199  2.21433
##
## 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    : (0.873, NA)
## delta  [Fix]   : 1.000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.147535   2.685655 -0.4273   0.6697
## x             0.106595   0.073089  1.4584   0.1466
##
## Residual standard error: 0.9448 on 163 degrees of freedom
## Multiple R-squared: 0.01288, Adjusted R-squared: 0.006825
## F-statistic: 2.127 on 1 and 163 DF,  p-value: 0.1466

```

```

# from MOTMOT
picModel

```

```

## $model
##
## Call:
## lm(formula = formula, data = pic.data, x = TRUE, y = TRUE)
##
## Coefficients:
##      x
## 0.1066
##

```

```
##
## $model.summary
##
## Call:
## lm(formula = formula, data = pic.data, x = TRUE, y = TRUE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1581 -0.8276 -0.1225  0.6007  2.3315
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## x   0.10660     0.07309   1.458   0.147
##
## Residual standard error: 0.9448 on 163 degrees of freedom
## Multiple R-squared:  0.01288,    Adjusted R-squared:  0.006825
## F-statistic: 2.127 on 1 and 163 DF,  p-value: 0.1466
##
##
## $intercept
## [1] -1.147535
##
## $lambda
## [1] 1
##
## $logLikelihood
##           [,1]
## [1,] -523.2653
##
## $AIC
##           [,1]
## [1,] 1047.531
```

## 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.)