

# Package ‘slouch’

October 14, 2022

**Type** Package

**Title** Stochastic Linear Ornstein-Uhlenbeck Comparative Hypotheses

**Version** 2.1.4

**Date** 2020-02-21

**Description** An implementation of a phylogenetic comparative method. It can fit univariate among-species Ornstein-Uhlenbeck models of phenotypic trait evolution, where the trait evolves towards a primary optimum. The optimum can be modelled as a single parameter, as multiple discrete regimes on the phylogenetic tree, and/or with continuous covariates. See also Hansen (1997) <[doi:10.2307/2411186](https://doi.org/10.2307/2411186)>, Butler & King (2004) <[doi:10.1086/426002](https://doi.org/10.1086/426002)>, Hansen et al. (2008) <[doi:10.1111/j.1558-5646.2008.00412.x](https://doi.org/10.1111/j.1558-5646.2008.00412.x)>.

**License** GPL-2

**LazyData** FALSE

**ByteCompile** FALSE

**RoxygenNote** 7.0.2

**Depends** R (>= 3.0.0), base, stats

**Imports** crayon, parallel, ape, memoise

**Suggests** knitr, rmarkdown, bookdown, testthat

**URL** <http://github.com/kopperud/slouch>

**BugReports** <http://github.com/kopperud/slouch/issues>

**VignetteBuilder** knitr

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Bjørn Tore Kopperud [aut, cre],  
Jason Pienaar [aut],  
Kjetil Lysne Voje [aut],  
Steven Hecht Orzack [aut],  
Thomas F. Hansen [aut],  
Mark Grabowski [ctb]

**Maintainer** Bjørn Tore Kopperud <kopperud@protonmail.com>

**Repository** CRAN

**Date/Publication** 2020-02-21 13:30:02 UTC

## R topics documented:

slouch-package . . . . .	2
artiodactyla . . . . .	3
brown.fit . . . . .	4
hillclimbplot . . . . .	7
logLik.slouch . . . . .	8
lseq . . . . .	9
neocortex . . . . .	9
plot.slouch . . . . .	10
print.slouch . . . . .	11
regimeplot.slouch . . . . .	12
slouch.fit . . . . .	13
summary.slouch . . . . .	16
<b>Index</b>	<b>18</b>

---

slouch-package	<i>SLOUCH: Stochastic Linear Ornstein Uhlenbeck Comparative Hypotheses</i>
----------------	--

---

## Description

An implementation of a phylogenetic comparative method. It can fit univariate among-species Ornstein-Uhlenbeck models of phenotypic trait evolution, where the trait evolves towards a primary optimum. The optimum can be modelled as a single parameter, as multiple discrete regimes on the phylogenetic tree, and/or with continuous covariates. See also Hansen (1997) doi:10.2307/2411186, Butler & King (2004) doi:10.1086/426002, Hansen et al. (2008) doi:10.1111/j.1558-5646.2008.00412.x.

## References

- Hansen, T. F. (1997). Stabilizing Selection and the Comparative Analysis of Adaptation. *Evolution*, 51(5), 1341. <https://doi.org/10.2307/2411186>
- Hansen, T. F., Pienaar, J., & Orzack, S. H. (2008). A comparative method for studying adaptation to a randomly evolving environment. *Evolution*, 62(8), 1965–1977. <https://doi.org/10.1111/j.1558-5646.2008.00412.x>
- Labra, A., Pienaar, J., & Hansen, T. F. (2009). Evolution of Thermal Physiology in *Liolaemus* Lizards: Adaptation, Phylogenetic Inertia, and Niche Tracking. *The American Naturalist*, 174(2), 204–220. <https://doi.org/10.1086/600088>
- Hansen, T. F., & Bartoszek, K. (2012). Interpreting the evolutionary regression: The interplay between observational and biological errors in phylogenetic comparative studies. *Systematic Biology*, 61(3), 413–425. <https://doi.org/10.1093/sysbio/syr122>

- Escudero, M., Hipp, A. L., Hansen, T. F., Voje, K. L., & Luceño, M. (2012). Selection and inertia in the evolution of holocentric chromosomes in sedges (Carex, Cyperaceae). *New Phytologist*, 195(1), 237–247. <https://doi.org/10.1111/j.1469-8137.2012.04137.x>

**Author(s)**

**Maintainer:** Bjørn Tore Kopperud <kopperud@protonmail.com>

Authors:

- Jason Pienaar
- Kjetil Lysne Voje
- Steven Hecht Orzack
- Thomas F. Hansen

Other contributors:

- Mark Grabowski [contributor]

**See Also**

Useful links:

- <http://github.com/kopperud/slouch>
- Report bugs at <http://github.com/kopperud/slouch/issues>

---

artiodactyla

*Artiodactyla Phylogeny*

---

**Description**

A phylogenetic tree of Artiodactyla (even-toed ungulates).

**Usage**

```
data(artiodactyla)
```

**Format**

An object of class "phylo", from package "ape".

**References**

The tree is taken from Toljagic et al. (2017).

- Toljagić, O., Voje, K. L., Matschiner, M., Liow, L. H., & Hansen, T. F. (2017). Millions of years behind: Slow adaptation of ruminants to grasslands. *Systematic Biology*, (318). <https://doi.org/10.1093/sysbio/syx059>

## Examples

```
data(artiodactyla)
library(ape)
plot(artiodactyla)

# Note: This tree also has regime information for each internal node in the tree:
print(artiodactyla$node.label)
```

---

brown.fit

*Function to fit Brownian-motion models of trait evolution*

---

## Description

Function to fit Brownian-motion models of trait evolution

## Usage

```
brown.fit(
  phy,
  species = NULL,
  sigma2_y_values = NULL,
  response,
  mv.response = NULL,
  fixed.fact = NULL,
  direct.cov = NULL,
  mv.direct.cov = NULL,
  mcov.direct.cov = NULL,
  random.cov = NULL,
  mv.random.cov = NULL,
  mcov.random.cov = NULL,
  estimate.Ya = FALSE,
  interactions = FALSE,
  hessian = FALSE,
  support = 2,
  convergence = 1e-06,
  nCores = 1,
  hillclimb = TRUE,
  lower = 1e-08,
  upper = NULL,
  verbose = FALSE
)
```

## Arguments

phy                    an object of class 'phylo', must be rooted.

species	a character vector of species tip labels, typically the "species" column in a data frame. This column needs to be an exact match and same order as phy\$tip.label
sigma2_y_values	a vector of one or more candidates for sigma squared (y) to be evaluated in grid search.
response	a numeric vector of a trait to be treated as response variable
mv.response	numeric vector of the observational variances of each response trait. E.g if response is a mean trait value, mv.response is the within-species squared standard error of the mean.
fixed.fact	factor of regimes on the terminal edges of the tree, in same order as species. If this is used, phy\$node.label needs to be filled with the corresponding internal node regimes, in the order of node indices (root: n+1),(n+2),(n+3), ...
direct.cov	Direct effect independent variables
mv.direct.cov	Estimation variances for direct effect independent variables. Must be the same shape as direct.cov
mcov.direct.cov	Estimation covariances between the response variable and direct effect independent variables. Must be the same shape as direct.cov
random.cov	Independent variables each modeled as a brownian motion
mv.random.cov	Estimation variances for the brownian covariates. Must be the same shape as random.cov
mcov.random.cov	Estimation covariances between the response variable and random effect independent variables. Must be the same shape as random.cov
estimate.Ya	independently estimates the ancestral state under Brownian motion. Note that, for an intercept model, the intercept IS the ancestral state estimate (since there are no directional or stabilizing trends in a standard Brownian motion).
interactions	a logical value. Whether to model interactions between (all) direct-effect continuous covariates and categorical regimes (experimental). Defaults to FALSE
hessian	use the approximate hessian matrix at the likelihood peak as found by the hill-climber, to compute standard errors for the parameters that enter in parameter search.
support	a scalar indicating the size of the support set, defaults to 2 units of log-likelihood.
convergence	threshold of iterative GLS estimation for when beta is considered to be converged.
nCores	number of CPU cores used in grid-search. If 2 or more cores are used, all print statements are silenced during grid search. If performance is critical it is recommended to compile and link R to a multithreaded BLAS, since most of the heavy computations are common matrix operations. Even if a singlethreaded BLAS is used, this may or may not improve performance, and performance may vary with OS.
hillclimb	logical, whether to use hillclimb parameter estimation routine or not. This routine (L-BFGS-B from optim()) may be combined with the grid-search, in which case it will on default start on the sigma and halflife for the local ML found by the grid-search.

lower	lower bounds for the optimization routine, defaults to 1e-8. When running direct effect models without observational error, it may be useful to specify a positive lower bounds for the sigma squared, since the residual variance-covariance matrix is degenerate when sigma = 0.
upper	upper bounds for the optimization routine, defaults to 10 * var(response) * max(treeheight).
verbose	a logical value indicating whether to print a summary in each iteration of parameter search. May be useful when diagnosing unexpected behaviour or crashes.

### Value

An object of class 'slouch', essentially a list with the following fields:

parameter_space	a list of the entire parameter space traversed by the grid search and the hill-climber as applicable.
tree	a list of parameters concerning the tree: <ul style="list-style-type: none"> <li>• phy - an object of class 'phy'</li> <li>• T.term - a numeric vector including the time from the root of the tree to the tip, for all taxa 1,2,3... n.</li> <li>• ta - for all pairs of species, the time from their most recent common ancestor (mrca) to the root of the tree.</li> <li>• tia - for all pairs of species, the time from their mrca to the tip of species i.</li> <li>• tja - the transpose of tia.</li> <li>• tij - for all pairs of species, the time from species i to their mrca, plus the time from their mrca to species j. In other words, tia + transpose(tia).</li> <li>• times - for all nodes (1,2,3... n, root, root+1, ...) in the tree, the time from the root to said node.</li> <li>• lineages - for all species (1,2,3... n), a list of their branch times and regimes as painted on the tree.</li> <li>• regimes - for all nodes (1,2,3... n, root, root+1, ...) in the tree, the respective regime as specified by "phy\$node.label" and "fixed.fact".</li> </ul>
modfit	a list of statistics to characterize model fit
supportplot	a list or matrix used to plot the grid search
supported_range	a matrix indicating the interval of grid search that is within the support region. If the grid search values are carefully selected, this may be used to estimate the true support region.
V	the residual variance-covariance matrix for the maximum likelihood model as found by parameter search.
evolpar	maximum likelihood estimates of parameters under the chosen model.
beta_primary	regression coefficients and associated objects. Whether the regression coefficients are to be interpreted as optima or not depend on the type of model and model estimates.

beta_evolutionary	under a random effect model, "beta_evolutionary" is the evolutionary regression coefficients and associated objects.
n.par	number of free parameters with which the likelihood criteria are penalized.
brownian_predictors	under a random effect model, a matrix of means and standard errors for the independent Brownian motion variable(s). Not to be confused with the regression coefficients when the residuals are under a "bm" model.
climblog_df	a matrix of the path trajectory of the hillclimber routine.
fixed.fact	the respective regimes for all species (1,2,3... n).
control	internal parameters for control flow.

---

hillclimbplot	<i>Plot the hillclimber trajectory</i>
---------------	--

---

## Description

Plot the hillclimber trajectory

## Usage

```
hillclimbplot(x, ...)

## S3 method for class 'slouch'
hillclimbplot(x, ...)
```

## Arguments

x	An object of class 'slouch'
...	Additional arguments passed to 'plot.default(...)'

## Methods (by class)

- slouch: Hillclimbplot for the 'slouch object'

## Examples

```
library(slouch)
library(ape)

data(neocortex)
data(artiodyctyla)

neocortex <- neocortex[match(artiodyctyla$tip.label, neocortex$species), ]

m0 <- slouch.fit(phy = artiodyctyla,
                 species = neocortex$species,
```

```

        response = neocortex$neocortex_area_mm2_log_mean,
        mv.response = neocortex$neocortex_se_squared,
        hillclimb = TRUE)

hillclimbplot(m0)

m1 <- brown.fit(phy = artiodactyla,
               species = neocortex$species,
               response = neocortex$neocortex_area_mm2_log_mean,
               mv.response = neocortex$neocortex_se_squared,
               hillclimb = TRUE)

hillclimbplot(m1)

```

---

logLik.slouch

*Extract Log-Likelihood*


---

### Description

Extract Log-Likelihood

### Usage

```

## S3 method for class 'slouch'
logLik(object, ...)

```

### Arguments

object	An object of class 'slouch'
...	Additional arguments.

### Value

An object of class 'logLik'

### Examples

```

data(artiodactyla)
data(neocortex)

neocortex <- neocortex[match(artiodactyla$tip.label, neocortex$species), ]

m0 <- slouch.fit(phy = artiodactyla,
                species = neocortex$species,
                response = neocortex$body_mass_g_log_mean,
                mv.response = neocortex$body_mass_g_log_varmean,
                fixed.fact = neocortex$diet,
                hillclimb = TRUE)

logLik(m0)

```



---

lseq	<i>Logarithmically spaced sequence</i>
------	--

---

**Description**

Logarithmically spaced sequence

**Usage**

```
lseq(from = 1, to = 1e+05, length.out = 6)
```

**Arguments**

from            the starting value of the sequence. Must be positive.  
to              the terminal value of the sequence. Must be larger than input to "from".  
length.out     desired length of the sequence. Must not be negative.

**Value**

A sequence of logarithmically spaced numbers.

**Examples**

```
lseq(1, 1000, length.out = 4)
```

---

neocortex	<i>Artiodactyl brain data</i>
-----------	-------------------------------

---

**Description**

Morphological data on mean neocortex area (mm<sup>2</sup>), mean brain size (g) and mean body size (g) for 43 species, including estimates of within-species observational error. These standard errors are not based directly on the unbiased sample variance estimator. The sample variances were first calculated, then a "global sample variance" was estimated using a weighted average. Lastly, the global sample variance was divided by the respective within-species sample size for each species to obtain the squared standard error as reported in the dataset.

**Usage**

```
data(neocortex)
```

**Format**

An object of class "data.frame".

## References

The following literature includes details on the original data collection.

- Haarmann, K., & Oboussier, H. (1972). Morphologische und quantitative Neocortexuntersuchungen bei Boviden, ein Beitrag zur Phylogenie dieser Familie. II. Formen geringen Körpergewichts (3kg - 25kg) aus den Subfamilien Cephalophinae und Antilopinae. Mitteilungen Aus Dem Hamburgischen Zoologischen Museum Und Institut, 68, 231–269.
- Oboussier, H. (1972). Morphologische und quantitative Neocortexuntersuchungen bei Boviden, ein Beitrag zur Phylogenie dieser Familie III. Formen über 75 kg Körpergewicht. Mitteilungen Aus Dem Hamburgischen Zoologischen Museum Und Institut, 68, 271–292.
- Oboussier, H. (1978). Zur Kenntnis des Bergnyalas (*Tragelaphus buxtoni*) und des Bongos (*Taurotragus euryceros*). Untersuchungen über den Körperbau und das Gehirn. Zeitschrift Für Säugetierkunde, 43, 114–125.
- Oboussier, H., & Möller, G. (1971). Zur Kenntnis des Gehirns der Giraffidae (Pecora, Artiodactyla, Mammalia) - ein Vergleich der Neocortex-Oberflächengröße. Zeitschrift Für Säugetierkunde, 36, 291–296.
- Ronnefeld, U. (1970). Morphologische und quantitative Neocortexuntersuchungen bei Boviden, ein Beitrag zur Phylogenie dieser Familie. I. Formen mittlerer Körpergröße (25 kg bis 75 kg). Gegenbaurs Morphologische Jahrbuch, 161–230.

See also:

- Haarmann, K. (1975). Morphological and histological study of neocortex of bovids (Antilopinae, Cephalophinae) and Tragulidae with comments on evolutionary development. Journal Fur Hirnforschung, 16, 93–116.
- Oboussier, H. (1979). Evolution of the brain and phylogenetic development of African Bovidae. South African Journal of Zoology, 14(3), 119–124. <https://doi.org/10.1080/02541858.1979.11447660>

## Examples

```
data(neocortex)
plot(neocortex$brain_mass_g_log_mean, neocortex$neocortex_area_mm2_log_mean)
```

---

plot.slouch

*Plot Grid Search*

---

## Description

Graphical plot of parameter space traversed by the grid search.

## Usage

```
## S3 method for class 'slouch'
plot(x, theta = 30, phi = 30, expand = 0.5, ltheta = 120, shade = 0.75, ...)
```

**Arguments**

x	An object of class 'slouch'
theta	angles defining the viewing direction. theta gives the azimuthal direction and phi the colatitude.
phi	angles defining the viewing direction. theta gives the azimuthal direction and phi the colatitude.
expand	a expansion factor applied to the z coordinates. Often used with $0 < \text{expand} < 1$ to shrink the plotting box in the z direction.
ltheta	if finite values are specified for ltheta and lphi, the surface is shaded as though it was being illuminated from the direction specified by azimuth ltheta and colatitude lphi.
shade	the shade at a surface facet is computed as $((1+d)/2)^{\text{shade}}$ , where d is the dot product of a unit vector normal to the facet and a unit vector in the direction of a light source. Values of shade close to one yield shading similar to a point light source model and values close to zero produce no shading. Values in the range 0.5 to 0.75 provide an approximation to daylight illumination.
...	Additional parameters passed to persp(...) or plot(...)

**Examples**

```

data(artiodyctyla)
data(neocortex)

neocortex <- neocortex[match(artiodyctyla$tip.label, neocortex$species), ]

m0 <- slouch.fit(phy = artiodyctyla,
                hl_values = seq(0.001, 50, length.out = 15),
                vy_values = seq(0.001, 3, length.out = 15),
                species = neocortex$species,
                response = neocortex$body_mass_g_log_mean,
                mv.response = neocortex$body_mass_g_log_varmean,
                fixed.fact = neocortex$diet)

plot(m0)

```

---

```
print.slouch          Print, minimalist output
```

---

**Description**

Print, minimalist output

**Usage**

```
## S3 method for class 'slouch'
print(x, ...)
```

**Arguments**

x                    An object of class 'slouch'  
 ...                  additional parameters

**Examples**

```
data(artiodactyla)
data(neocortex)

neocortex <- neocortex[match(artiodactyla$tip.label, neocortex$species), ]

m0 <- slouch.fit(phy = artiodactyla,
                 hl_values = seq(0.001, 4, length.out = 10),
                 vy_values = seq(0.001, 0.05, length.out = 10),
                 species = neocortex$species,
                 response = neocortex$neocortex_area_mm2_log_mean,
                 mv.response = neocortex$neocortex_se_squared,
                 random.cov = neocortex$brain_mass_g_log_mean,
                 mv.random.cov = neocortex$brain_se_squared,
                 fixed.fact = neocortex$diet,
                 hillclimb = FALSE)
```

---

regimeplot.slouch      *Plot the internal regimes for a given fitted model*

---

**Description**

Plot the internal regimes for a given fitted model

**Usage**

```
## S3 method for class 'slouch'
regimeplot(x, ...)

regimeplot(x, ...)
```

**Arguments**

x                    an object of class 'slouch'  
 ...                  additional parameters passed to plot.phylo(...)

**Value**

nothing

**Methods (by class)**

- slouch: Regimeplot for the 'slouch' object

## Examples

```
data(artiodyctyla)
data(neocortex)

neocortex <- neocortex[match(artiodyctyla$tip.label, neocortex$species), ]

m0 <- slouch.fit(phy = artiodyctyla,
                 species = neocortex$species,
                 response = neocortex$body_mass_g_log_mean,
                 mv.response = neocortex$body_mass_g_log_varmean,
                 fixed.fact = neocortex$diet,
                 hillclimb = TRUE)

regimeplot(m0)
```

---

slouch.fit

*Function to fit Ornstein-Uhlenbeck models of trait evolution*

---

## Description

Function to fit Ornstein-Uhlenbeck models of trait evolution

## Usage

```
slouch.fit(
  phy,
  species = NULL,
  hl_values = NULL,
  a_values = NULL,
  vy_values = NULL,
  sigma2_y_values = NULL,
  response,
  mv.response = NULL,
  fixed.fact = NULL,
  direct.cov = NULL,
  mv.direct.cov = NULL,
  mcov.direct.cov = NULL,
  random.cov = NULL,
  mv.random.cov = NULL,
  mcov.random.cov = NULL,
  estimate.Ya = FALSE,
  estimate.bXa = FALSE,
  interactions = FALSE,
  hessian = FALSE,
  support = 2,
  convergence = 1e-06,
```

```

nCores = 1,
hillclimb = TRUE,
lower = c(1e-08, 1e-08),
upper = Inf,
verbose = FALSE
)

```

## Arguments

phy	an object of class 'phylo', must be rooted.
species	a character vector of species tip labels, typically the "species" column in a data frame. This column needs to be an exact match and same order as phy\$tip.label
hl_values	a vector of candidate phylogenetic half-life values to be evaluated in grid search. Optional.
a_values	a vector of candidate rate of adaptation values to be evaluated in grid search. Optional.
vy_values	a vector of candidate stationary variances for the response trait, to be evaluated in grid search. Optional.
sigma2_y_values	alternative to vy_values, if the stationary variance is reparameterized as the variance parameter for the Brownian motion.
response	a numeric vector of a trait to be treated as response variable
mv.response	numeric vector of the observational variances of each response trait. E.g if response is a mean trait value, mv.response is the within-species squared standard error of the mean.
fixed.fact	factor of regimes on the terminal edges of the tree, in same order as species. If this is used, phy\$node.label needs to be filled with the corresponding internal node regimes, in the order of node indices (root: n+1),(n+2),(n+3), ...
direct.cov	Direct effect independent variables
mv.direct.cov	Estimation variances for direct effect independent variables. Must be the same shape as direct.cov
mcov.direct.cov	Estimation covariances between the response variable and direct effect independent variables. Must be the same shape as direct.cov
random.cov	Independent variables each modeled as a brownian motion
mv.random.cov	Estimation variances for the brownian covariates. Must be the same shape as random.cov
mcov.random.cov	Estimation covariances between the response variable and random effect independent variables. Must be the same shape as random.cov
estimate.Ya	a logical value indicating whether "Ya" should be estimated. If true, the intercept $K = 1$ is expanded to $Y_a = \exp(-a*t)$ and $b_0 = 1 - \exp(-a*t)$ . If models with categorical covariates are used, this will instead estimate a separate primary optimum for the root niche, "Ya". This only makes sense for non-ultrametric trees. If the tree is ultrametric, the model matrix becomes singular.

estimate.bXa	a logical value indicating whether "bXa" should be estimated. If true, $bXa = 1 - \exp(-a*t) - (1 - (1 - \exp(-a*t))/(a*t))$ is added to the model matrix, estimating $b*Xa$ . Same requirements as for estimating $Y_a$ .
interactions	a logical value. Whether to model interactions between (all) direct-effect continuous covariates and categorical regimes (experimental). Defaults to FALSE
hessian	use the approximate hessian matrix at the likelihood peak as found by the hill-climber, to compute standard errors for the parameters that enter in parameter search.
support	a scalar indicating the size of the support set, defaults to 2 units of log-likelihood.
convergence	threshold of iterative GLS estimation for when beta is considered to be converged.
nCores	number of CPU cores used in grid-search. If 2 or more cores are used, all print statements are silenced during grid search. If performance is critical it is recommended to compile and link R to a multithreaded BLAS, since most of the heavy computations are common matrix operations. Even if a singlethreaded BLAS is used, this may or may not improve performance, and performance may vary with OS.
hillclimb	logical, whether to use hillclimb parameter estimation routine or not. This routine (L-BFGS-B from <code>optim()</code> ) may be combined with the grid-search, in which case it will on default start on the sigma and halflife for the local ML found by the grid-search.
lower	lower bounds for the optimization routine, defaults to <code>c(0,0)</code> . First entry in vector is half-life, second is stationary variance. When running direct effect models without observational error, it may be useful to specify a positive lower bounds for the stationary variance, e.g <code>c(0, 0.001)</code> , since the residual variance-covariance matrix is degenerate when $\sigma = 0$ .
upper	upper bounds for the optimization routine, defaults to <code>c(Inf, Inf)</code> .
verbose	a logical value indicating whether to print a summary in each iteration of parameter search. May be useful when diagnosing unexpected behaviour or crashes.

### Value

An object of class 'slouch', essentially a list with the following fields:

parameter_space	a list of the entire parameter space traversed by the grid search and the hill-climber as applicable.
tree	a list of parameters concerning the tree: <ul style="list-style-type: none"> <li>• phy - an object of class 'phy'</li> <li>• T.term - a numeric vector including the time from the root of the tree to the tip, for all taxa 1,2,3... n.</li> <li>• ta - for all pairs of species, the time from their most recent common ancestor (mrca) to the root of the tree.</li> <li>• tia - for all pairs of species, the time from their mrca to the tip of species i.</li> <li>• tja - the transpose of tia.</li> </ul>

- tij - for all pairs of species, the time from species i to their mrca, plus the time from their mrca to species j. In other words, tia + transpose(tia).
- times - for all nodes (1,2,3... n, root, root+1, ...) in the tree, the time from the root to said node.
- lineages - for all species (1,2,3... n), a list of their branch times and regimes as painted on the tree.
- regimes - for all nodes (1,2,3... n, root, root+1, ...) in the tree, the respective regime as specified by "phy\$node.label" and "fixed.fact".

modfit	a list of statistics to characterize model fit
supportplot	a list or matrix used to plot the grid search
supported_range	a matrix indicating the interval of grid search that is within the support region. If the grid search values are carefully selected, this may be used to estimate the true support region.
V	the residual variance-covariance matrix for the maximum likelihood model as found by parameter search.
evolpar	maximum likelihood estimates of parameters under the chosen model.
beta_primary	regression coefficients and associated objects. Whether the regression coefficients are to be interpreted as optima or not depend on the type of model and model estimates.
beta_evolutionary	under a random effect model, "beta_evolutionary" is the evolutionary regression coefficients and associated objects.
n.par	number of free parameters with which the likelihood criteria are penalized.
brownian_predictors	under a random effect model, a matrix of means and standard errors for the independent Brownian motion variable(s). Not to be confused with the regression coefficients when the residuals are under a "bm" model.
climblog_df	a matrix of the path trajectory of the hillclimber routine.
fixed.fact	the respective regimes for all species (1,2,3... n).
control	internal parameters for control flow.

---

summary.slouch

*Model Summary*


---

## Description

Model Summary

## Usage

```
## S3 method for class 'slouch'
summary(object, ...)
```



**Arguments**

object	An object of class 'slouch'
...	Additional arguments, unused.

**Examples**

```
data(artiodactyla)
data(neocortex)

neocortex <- neocortex[match(artiodactyla$tip.label, neocortex$species), ]

m0 <- slouch.fit(phy = artiodactyla,
  hl_values = seq(0.001, 4, length.out = 10),
  vy_values = seq(0.001, 0.05, length.out = 10),
  species = neocortex$species,
  response = neocortex$neocortex_area_mm2_log_mean,
  mv.response = neocortex$neocortex_se_squared,
  random.cov = neocortex$brain_mass_g_log_mean,
  mv.random.cov = neocortex$brain_se_squared,
  fixed.fact = neocortex$diet,
  hillclimb = FALSE)

summary(m0)

plot(m0, theta = 150)
```

# Index

## \* datasets

- artiodactyla, [3](#)
- neocortex, [9](#)

artiodactyla, [3](#)

brown.fit, [4](#)

hillclimbplot, [7](#)

logLik.slouch, [8](#)

lseq, [9](#)

neocortex, [9](#)

plot.slouch, [10](#)

print.slouch, [11](#)

regimeplot (regimeplot.slouch), [12](#)

regimeplot.slouch, [12](#)

slouch (slouch-package), [2](#)

slouch-package, [2](#)

slouch.fit, [13](#)

summary.slouch, [16](#)