The paleoTSalt Package

October 23, 2007

Type Package

Title Modeling evolution in paleontological time-series (alternate parameterization)

Version 0.1-1

Date 2007-10-22

Author Gene Hunt <hunte@si.edu>

Maintainer Gene Hunt <hunte@si.edu>

Depends paleoTS

Description This package facilitates analysis of paleontological sequences of trait values from an evolving lineage. Functions are provided to fit, using maximum likelihood, evolutionary models including unbiased random walks, directional evolution, stasis and Ornstein-Uhlenbeck (OU) models. This package performs many of the same functions as the package paleoTS, but does so using a different parameterization of the evolutionary models.

License GPL version 2 or newer

R topics documented:

add.OU.curves .......................... 2
Stickleback data .......................... 3
fit3models.alt .......................... 4
logL.alt.GRW .......................... 5
opt.alt.GRW .......................... 6
paleoTSalt-package .......................... 8
Stickleback data .......................... 9
Stickleback data .......................... 10
sim.OU .......................... 11

Index 13
add.OU.curves

Adds curves to an existing plot indicating best-fit OU model

Description

This function takes the results of fitting the Ornstein-Uhlenbeck (OU) model and plots the expected value (mean) and 95 percent probability interval of the best-fitting OU parameters.

Usage

```r
add.OU.curves(w, x, what = c("lines", "polygon"), tt.offset = 0, ...)
```

Arguments

- `w`: output from the function `opt.alt.OU`
- `x`: a `paleoTS` object
- `what`: whether to draw lines or polygon for the model fit
- `tt.offset`: difference between starting age in `w` and `x`
- `...`: additional arguments to graphing functions

Details

Note that `tt.offset` is only used in the relatively rare instances in which the model fit is not applied to the entire sequence.

Value

None, the function is evoked to add graphics to an existing plot.

Author(s)

Gene Hunt

See Also

- `plot.paleoTS, opt.alt.OU`

Examples

```r
data(dorsal.spines)
# get subset of samples from invading lineage (tt>=4.5 Kyr), only those with nn>=5
ok <- dorsal.spines$tt >= 4.5 & dorsal.spines$nn >=5
ds2 <- sub.paleoTS(dorsal.spines, ok=ok)

# convert time scale to generations (500 gen per Kyr)
ds2$tt <- ds2$tt*(1000/2)
plot(ds2, pool=TRUE)
```
Stickleback data

# fit OU model
m.ou<- opt.alt.OU(ds2, pool=TRUE)
add.OU.curves(m.ou, ds2, what="lines", col="darkgrey")

Description

Data of dorsal spine data from a fossil stickleback lineage.

Usage

data(dorsal.spines)

Format

a paleoTS object for the time-series of dorsal spine counts

Details

These data are already in the form of a paleoTS object, with sample means (mm), variances (vv) sample sizes (nn), and ages in Kyr (tt).

Source


Examples

data(dorsal.spines)

# get subset of samples from invading lineage (tt>=4.5 Kyr), only those with nn>=5
ok<- dorsal.spines$tt >= 4.5 & dorsal.spines$nn >=5
ds2<- sub.paleoTS(dorsal.spines, ok=ok)

# convert time scale to generations (500 gen per Kyr)
ds2$tt<- ds2$tt*(1000/2)
plot(ds2, pool=TRUE)

# compare URW (drift) and OU (adaptive) models
m.urw<- opt.alt.URW(ds2, pool=TRUE)
m.ou<- opt.alt.OU(ds2, pool=TRUE)
aw<- akaike.wts(c(m.urw$AICc, m.ou$AICc))
cat("Akaike wts: ", round(aw, 3),"\t", "[URW, OU]", "\n")
fit3models.alt  

Do model fits for three evolutionary models (alternate parameterization)

Description

Function fits three models to an evolutionary time series: (1) general random walk, (2) unbiased random walk, and (3) stasis.

Usage

fit3models.alt(y, pool = TRUE, silent = FALSE, wts = "AICc")

Arguments

- **y**: a paleoTS object
- **pool**: logical, if TRUE, variances are pooled across samples
- **silent**: logical, if TRUE, results are not printed
- **wts**: what version of AIC to use for calculation of Akaike weights; either AIC or AICc

Value

If silent=FALSE, function fit3models.alt returns a vector of parameter estimates; if silent=TRUE, a list with the following components is returned, with models listed in the order of general random walk, unbiased random walk, and stasis:

- **aic**: Akaike information criterion
- **aicc**: bias-corrected AIC
- **logl**: log-likelihoods of the three models
- **hats**: parameter estimates for general random walk (mstep, vstep), unbiased random walk (vstep) and stasis (theta, omega) models
- **ak.wts**: vector of Akaike weights

Warning

This function uses the alternate parameterization, and so its output is not comparable to those from paleoTS package.

Author(s)

Gene Hunt

References

logL.alt.GRW

See Also

opt.alt.GRW, fit3models

Examples

```r
x1 <- sim.GRW(ns=30, ms=0.5, vs=0.2)
fit3models.alt(x1)
x2 <- sim.Stasis(ns=30, theta=10, omega=0)
fit3models.alt(x2)
```

Description

Returns log-likelihood for general random walk (logL.alt.GRW), unbiased random walk (logL.alt.URW), stasis (logL.alt.Stasis) and OU (logL.alt) models.

Usage

```r
logL.alt.GRW(p, x)
logL.alt.URW(p, x)
logL.alt.Stasis(p, x)
logL.alt.OU(p, x)
```

Arguments

- `p`: a vector of parameters
- `x`: a paleoTS object

Details

For the general random walk, `p = c(anc, mstep, vstep)`; for an unbiased random walk, `p = c(anc, vstep)`; for the stasis model, `p = c(theta, omega)`, and for the OU model `p = c(anc, vstep, theta, alpha)`. In general, users will not be access these functions directly, but instead use the optimization functions, which use these functions to find the best-supported parameter values.

Value

The log-likelihood of the parameter estimates (p), given the data (x).

Warning

Because these functions parameterize the models differently, their log-likelihoods are not comparable to those from paleoTS package.
Author(s)

Gene Hunt

References


See Also

opt.alt.GRW, logL.GRW

Examples

```r
x <- sim.GRW(ns=20, ms=0, vs=1)
L1 <- logL.alt.GRW(p=c(0,0,1), x)                         # actual parameters
L2 <- logL.alt.GRW(p=c(0,10,1), x)                         # should be a bad guess
cat(L1, L2, "\n")
```

opt.alt.GRW

*Optimize evolutionary models (alternate parameterization)*

Description

Functions to find maximum likelihood solutions to general random walk (opt.alt.GRW), unbiased random walk (opt.alt.URW), stasis (opt.alt.Stasis) and OU models (opt.alt.OU).

Usage

```r
opt.alt.GRW(x, pool = TRUE, cl = list(fnscale = -1), meth = "L-BFGS-B", hess = FALSE)
opt.alt.URW(x, pool = TRUE, cl = list(fnscale = -1), meth = "L-BFGS-B", hess = FALSE)
opt.alt.Stasis(x, pool = TRUE, cl = list(fnscale = -1), meth = "L-BFGS-B", hess = FALSE)
opt.alt.OU(x, pool = TRUE, cl = list(fnscale = -1), meth = "L-BFGS-B", hess = FALSE)
```

Arguments

- `x`: a paleoTS object
- `pool`: logical indicating whether to pool variances across samples
- `cl`: control list, passed to function optim
- `meth`: optimization method, passed to function optim
- `hess`: logical, indicating whether to calculate standard errors from the Hessian matrix
Details

These functions numerically search a log-likelihood surface for its optimum—they are a convenient wrapper to `optim`. Arguments `meth`, `cl`, and `hess` are passed to `optim`; see the help for that function for details. These are included to allow sophisticated users greater control over the optimization; the defaults seem to work well for most, but not all sequences. For `meth="L-BFGS-B"`, some parameters are constrained to be non-negative, which is useful parameters which cannot truly be negative, such as `vstep` (random walk) and `omega` (stasis model).

Initial estimates to start the optimization come in part from analytical solutions based on assuming equal sampling error across samples and evenly spaced samples in time (functions `mle.GRW`, `mle.URW` and `mle.Stasis`).

Value

- `par`: parameter estimates
- `value`: the log-likelihood of the optimal solution
- `counts`: returned by `optim`
- `convergence`: returned by `optim`
- `message`: returned by `optim`
- `p0`: initial guess for parameter values at start of optimization
- `K`: number of parameters in the model
- `n`: the number of observations, equal to the number of samples
- `AIC`: Akaike information criterion
- `AICc`: bias-corrected Akaike information criterion
- `BIC`: Bayes (or Schwarz) information criterion
- `se`: standard errors for parameter estimates, computed from the curvature of the log-likelihood surface (only if `hess = TRUE`)
- `...`: other output from call to `optim`

Warning

Because these functions parameterize the models differently, their outputs are not comparable to those from `paleoTS` package.

Note

These optimizations are performed with a parameterization of the GRW, URW and Stasis models that differs from the package `paleoTS` (the OU model is not implemented in that package). Specifically, in `paleoTS`, models are fit from the differences between adjacent samples, removing the autocorrelation in the time-series. In package paleoTSalt, models are fit using the actual sample values, with the autocorrelation among samples accounted for in the log-likelihood function. With each model, the joint distribution of sample means is multivariate normal, with means and variance-covariances determined by evolutionary parameters and sampling errors.

For details on this parameterization of the models, see the upcoming paper in Evolution by G. Hunt, M. Bell and M. Travis.
paleoTSalt-package

Author(s)

Gene Hunt

References


See Also

`logL.alt.GRW`, `opt.GRW`

Examples

```r
x <- sim.GRW(ns=30, ms=1, vs=1)
plot(x)
m.urw <- opt.alt.URW(x)
m.grw <- opt.alt.GRW(x)
m.sta <- opt.alt.Stasis(x)

cat(m.urw$AICc, m.grw$AICc, m.sta$AICc, "\n")  # print AICc scores
```

paleoTSalt-package  *Fitting models of evolution to paleontological time-series of phenotypic data*

Description

This package facilitates analysis of paleontological sequences of trait values from an evolving lineage. Functions are provided to fit, using maximum likelihood, evolutionary models including unbiased random walks, directional evolution, stasis and OU (Orstein-Uhlenbeck) models. This package has a similar purpose as the *paleoTS* package, but employs a different parameterization of evolutionary models.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>paleoTSalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.1-1</td>
</tr>
<tr>
<td>Date</td>
<td>2007-10-22</td>
</tr>
<tr>
<td>License</td>
<td>GPL 3 or newer</td>
</tr>
</tbody>
</table>
**Stickleback data**

**Author(s)**

Gene Hunt, ⟨hunte@si.edu⟩

**References**


**See Also**

paleoTS-package

**Examples**

```r
data(dorsal.spines)

# get subset of samples from invading lineage (tt>=4.5 Kyr), only those with nn>=5
ok< - dorsal.spines$tt >= 4.5 & dorsal.spines$nn >=5
ds2< - sub.paleoTS(dorsal.spines, ok=ok)

# convert time scale to generations (500 gen per Kyr)
ds2$t< - ds2$t*(1000/2)
plot(ds2, pool=TRUE)

# compare URW (drift) and OU (adaptive) models
m.urw< - opt.alt.URW(ds2, pool=TRUE)
m.ou< - opt.alt.OU(ds2, pool=TRUE)
cat ("Model", "logL", "AICc\n", sep="|t")
cat ("URW", round(m.urw$val, 2), round(m.urw$AICc, 2), sep="|t")
cat ("OU", round(m.ou$val, 2), round(m.ou$AICc, 2), sep="|t")
add.OU.curves(m.ou, ds2, col="grey")
```

---

**Stickleback data**

**Stickleback data from Bell et al. (2006)**

**Description**

Data of pelvic score data from a fossil stickleback lineage.

**Usage**

data(dorsal.spines)

**Format**

a paleoTS object for the time-series of pelvic scores
Stickleback data

Details

These data are already in the form of a paleoTS object, with sample means (mm), variances (vv), sample sizes (nn), and ages in Kyr (tt).

Source


Examples

data(pelvic.score)

# get subset of samples from invading lineage (tt>=4.5 Kyr), only those with nn>=5
ok<- pelvic.score$tt >= 4.5 & pelvic.score$nn >=5
ps2<- sub.paleoTS(pelvic.score, ok=ok)

# convert time scale to generations (500 gen per Kyr)
ps2$tt<- ps2$tt*(1000/2)

plot(ps2, pool=TRUE)

---

Stickleback data Stickleback data from Bell et al. (2006)

Description

Counts of the number of touching pterygiophores from a fossil stickleback lineage.

Usage

data(dorsal.spines)

Format

a paleoTS object for the time-series of pterygiophore data

Details

These data are already in the form of a paleoTS object, with sample means (mm), variances (vv), sample sizes (nn), and ages in Kyr (tt).

Source

Examples

data(ptyeriophores)

# get subset of samples from invading lineage (tt>=4.5 Kyr), only those with nn>=5
ok<- pterygiophores$tt >= 4.5 & pterygiophores$nn >=5
pt2<- sub.paleoTS(ptyeriophores, ok=ok)

# convert time scale to generations (500 gen per Kyr)
pt2$tt<- pt2$tt*(1000/2)
plot(pt2, pool=TRUE)

# compare URW (drift) and OU (adaptive) models
m.urw<- opt.alt.URW(pt2, pool=TRUE)
m.ou<- opt.alt.OU(pt2, pool=TRUE)
aw<- akaike.wts(c(m.urw$AICc, m.ou$AICc))
cat("Akaike wts: ", round(aw, 3),"\t", "[URW, OU]", "\n")

sim.OU

Simulate evolutionary time-series

description
Generates an evolutionary time-series according to an Orstein-Uhlenbeck (OU) model.

Usage

sim.OU(ns = 20, anc = 0, theta = 10, alpha = 0.3, vs = 0.1, vp = 1, nn = rep(20, ns),
ou.M(anc, theta, aa, tt)
ou.V(vs, aa, tt)

Arguments

ns number of samples in time-series
anc ancestral phenotype at the start of the series
theta phenotype of the evolutionary optimum
alpha strength of the attracting force pulling the population to the optimum
vs step variance of the random walk component of change
vp within-population trait variance
nn vector of the number of individuals in each sample
tt vector of sample ages, increases from oldest to youngest
aa strength of the attracting force pulling the population to the optimum (same as alpha)
Details

See Hansen (1997) for a description of this model in a macroevolutionary context. This model also arises naturally in microevolution as a finite population evolving in the vicinity of an optimum in the adaptive landscape; see Lande (1976) and Estes & Arnold (2007).

Functions `ou.M` and `ou.V` are used internally by `sim.OU` and `add.OU.curves` to generate the means and variances of an OU process.

Value

A `paleoTS` object for `sim.OU`. For `ou.M` and `ou.V`, a vector of means or variances, respectively, are generated.

Author(s)

Gene Hunt

References


See Also

`sim.GRW`, `opt.alt.OU`

Examples

```r
x1 <- sim.OU(ns=100, anc=0, theta=10, alpha=0.2, vs=0.1, vp=0.1, nn=rep(100, times=100), tt=0)
plot(x1)
```
**Index**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Data/Models/Package</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>datasets</strong></td>
<td>Stickleback data, 3, 9, 10</td>
</tr>
<tr>
<td><strong>models</strong></td>
<td>add.OU.curves, 1</td>
</tr>
<tr>
<td></td>
<td>fit3models.alt, 4</td>
</tr>
<tr>
<td></td>
<td>logL.alt.GRW, 5</td>
</tr>
<tr>
<td></td>
<td>opt.alt.GRW, 6</td>
</tr>
<tr>
<td></td>
<td>sim.OU, 11</td>
</tr>
<tr>
<td><strong>package</strong></td>
<td>paleoTSalt-package, 8</td>
</tr>
<tr>
<td><strong>ts</strong></td>
<td>add.OU.curves, 1</td>
</tr>
<tr>
<td></td>
<td>fit3models.alt, 4</td>
</tr>
<tr>
<td></td>
<td>logL.alt.GRW, 5</td>
</tr>
<tr>
<td></td>
<td>opt.alt.GRW, 6</td>
</tr>
<tr>
<td></td>
<td>sim.OU, 11</td>
</tr>
<tr>
<td>add.OU.curves, 1</td>
<td></td>
</tr>
<tr>
<td>dorsal.spines (Stickleback data), 3</td>
<td></td>
</tr>
<tr>
<td>fit3models, 5</td>
<td></td>
</tr>
<tr>
<td>fit3models.alt, 4</td>
<td></td>
</tr>
<tr>
<td>logL.alt.GRW, 5, 8</td>
<td></td>
</tr>
<tr>
<td>logL.alt.OU(logL.alt.GRW), 5</td>
<td></td>
</tr>
<tr>
<td>logL.alt.Stasis(logL.alt.GRW), 5</td>
<td></td>
</tr>
<tr>
<td>logL.alt.URW(logL.alt.GRW), 5</td>
<td></td>
</tr>
<tr>
<td>logL.GRW, 6</td>
<td></td>
</tr>
<tr>
<td>opt.alt.GRW, 5, 6, 6</td>
<td></td>
</tr>
<tr>
<td>opt.alt.OU, 2, 12</td>
<td></td>
</tr>
<tr>
<td>opt.alt.OU(opt.alt.GRW), 6</td>
<td></td>
</tr>
<tr>
<td>opt.alt.Stasis(opt.alt.GRW), 6</td>
<td></td>
</tr>
<tr>
<td>opt.alt.URW(opt.alt.GRW), 6</td>
<td></td>
</tr>
<tr>
<td>opt.GRW, 8</td>
<td></td>
</tr>
<tr>
<td>ou.M(sim.OU), 11</td>
<td></td>
</tr>
<tr>
<td>ou.V(sim.OU), 11</td>
<td></td>
</tr>
<tr>
<td>paleoTS-package, 9</td>
<td></td>
</tr>
<tr>
<td>paleoTSalt (paleoTSalt-package), 8</td>
<td></td>
</tr>
<tr>
<td>paleoTSalt-package, 8</td>
<td></td>
</tr>
<tr>
<td>pelvic.score (Stickleback data), 9</td>
<td></td>
</tr>
<tr>
<td>plot.paleoTS, 2</td>
<td></td>
</tr>
<tr>
<td>pterygiophores (Stickleback data), 10</td>
<td></td>
</tr>
<tr>
<td>sim.GRW, 12</td>
<td></td>
</tr>
<tr>
<td>sim.OU, 11</td>
<td></td>
</tr>
<tr>
<td>Stickleback data, 3, 9, 10</td>
<td></td>
</tr>
</tbody>
</table>