BayesTraits Wrapper (BTW)

Oct 8, 2012

Author Randi Griffin <rgriff23@gmail.com>

Description This set of functions allows one to run BayesTraits from within R, conduct model comparisons and plot output from models.

Index

`MultistateML` 2

`MultistateMCMC` 2

`DiscreteML` 2

`DiscreteMCMC` 2

`ContinuousML` 2

`ContinuousMCMC` 2

`ContinuousRegressionML` 2

`ContinuousRegressionMCMC` 2

`lr.test` 6

`bf.test` 6

`plot.rates` 7

`plot.btmcmc` 8

`kill.bt` 9

Loading the functions

Use ‘setwd’ to navigate to the ‘BTW’ directory and enter this block of code into R:

```
for (n in 1:length(list.files('./R'))) {
      source(paste("./R/", list.files('./R')[n], sep=""))
}
```

The btw functions should now be loaded in your R session. To check that the functions are loaded, type `ls()` and the names of the functions should print to the console. If this step fails, ensure that ‘BTW’ is your current working directory and that the subdirectory ‘R’ is located inside ‘BTW’ and contains only R source files.

Run BayesTraits from R

Description

Run BayesTraits from R. By default, these functions search for BayesTraits in your current working directory.

Usage

```
MultistateML(tree, data, mlt = 10, res, resall, mrca, fo, et, silent =
TRUE)

MultistateMCMC(tree, data, res, resall, mrca, fo, et, it = 10000, bi =
1000, sa = 100, rd = 2, pr, hp, pa = "uniform 0 100", hpall, rj, rjhp,
silent = TRUE)

DiscreteML(tree, data, dependent = FALSE, mlt = 10, res, resall, mrca,
fo, et, silent=TRUE)

DiscreteMCMC(tree, data, dependent = FALSE, res, resall, mrca, fo, et,
it = 10000, bi = 1000, sa = 100, rd = 2, pr, hp, pa = "uniform 0 100",
hpall, rj, rjhp, silent = TRUE)

ContinuousML(tree, data, directional = FALSE, lambda = 1, kappa = 1,
delta = 1, mlt = 10)

ContinuousMCMC(tree, data, directional = FALSE, lambda = 1, kappa = 1,
delta = 1, it = 10000, bi = 1000, sa = 100, rd = 2, silent = TRUE)

ContinuousRegressionML(tree, data, lambda = 1, kappa = 1, delta = 1,
mlt = 10)

ContinuousRegressionMCMC(tree, data, lambda = 1, kappa = 1, delta = 1,
it = 10000, bi = 1000, sa = 100, rd = 2, silent = TRUE)
```

Arguments

`tree` an object of class `phylo` or `multiPhylo`.

`data` an object of class `data.frame`. The first column contains species names and remaining columns contain only the comparative data that will be used in the analysis (extraneous variables must be removed from the data frame). For continuous regression, the first data column is the response and remaining columns are predictor variables.

`silent` logical indicating whether to print BayesTraits output. Default is `FALSE`.

ML methods only

`mlt` integer specifying number of runs of the optimization algorithm. Default is 10.

MCMC methods only

`it` integer specifying number of MCMC iterations. Default is 10000.

`bi` integer specifying number of iterations to discard as burn-in. Default is 1000.

`sa` integer specifying number of iterations to skip between samples. Default is 100.

`rd` positive number specifying the rate deviation parameter. Default is 2.

Multistate and Discrete methods only

`res` character or vector indicating restrictions to place on rates. If a vector is given, each element indicates an independent restriction. Individual restrictions are given as character strings with the name of each rate separated by a single space, and the last set of characters in the string corresponds to either a rate or a non-negative number to restrict the other rates to. For example, if `res = c("q01 q10 1.2", "q12 q21")`, then rates q01 and q10 will be restricted to equal 1.2, and rate q12 will be restricted to equal q21.

`resall` character indicating a rate or a non-negative number to restrict all rates to.

`mrca` character or vector indicating nodes to reconstruct using the most recent common ancestor approach. If a vector is given, each element is a character string naming a pair of species whose most recent common ancestor will be reconstructed. Species names must be separated by a single space, e.g., `mrca = c("Lemur_catta Homo_sapiens")`.

`fo` character or vector indicating nodes to fix at particular states. If a vector is given, each element is a character string beginning with the state to be fixed, followed by the names of two species whose most recent common ancestor is represented by the node, all separated by a single space. For example, `fo = c("1 Lemur_catta Homo_sapiens", "2 Pan_troglodytes Homo_sapiens")` fixes the ancestral primate (mrca of Lemur and Homo) at state "1" and the ancestral great ape (mrca of Pan and Homo) at state "2".

`et` character or vector listing taxa to exclude.

Discrete methods only

`Dependent` logical indicating whether to run the dependent model of evolution (correlated evolution). Default is `FALSE`.

MultistateMCMC and DiscreteMCMC methods only `pr` character or vector describing prior distributions for model parameters. If a vector is given, each element is a character string specifying the distribution for a separate parameter by listing first the name of the parameter, then the name of the distribution (exp, gamma, uniform, beta), and finally the parameters that define the distribution. For example, `pr = c(“q01 exp 10”, “q10 gamma 10 10”)` sets an exponential distribution with mean 10 as the prior for rate q01, and a gamma distribution with mean and shape 10 as the prior for rate q10. By default, parameters have a uniform prior from -100 to 100.

`pa` character string specifying the prior distribution for all parameters by listing first the name of the parameter, then the name of the distribution (exp, gamma, uniform, beta), and finally the parameters that define the distribution. For example, `pa = “q01 exp 10”` sets an exponential distribution with mean 10 as the prior for all rates in the model.

`rj` toggles reversible jump model if non-empty. A character string specifying the prior distribution and it’s parameter(s) to use for the reversible jump model. For example, `rj = “exp 10”` implements a reversible jump model with an exponential prior of mean 10.

`rjhp` toggles reversible jump model with a hyper-prior if non-empty. A character string specifying the prior distribution and it’s parameter(s) to use for the reversible jump model, along with a minimum and maximum value for a uniform hyper-prior. For example, `rjhp = “exp 10 0 10”` implements a reversible jump model with an exponential prior of mean 10 and a hyper-prior from 0 to 10.

`hp` character or vector describing prior distributions and hyper-priors for model parameters. If a vector is given, each element is a character string specifying the prior and uniform hyper-prior for a separate parameter by listing the name of the parameter, then the name of the distribution (exp, gamma, uniform, beta), then the parameters that define the distribution, and finally the minimum and maximum of the hyper-prior. For example, `hp = “q01 exp 10 0 10”` sets an exponential distribution with mean 10 as the prior for rate q01, and a hyper-prior from 0 to 10 to seed the mean of the exponential prior.

`hpall` character string specifying the prior distribution and hyper-prior for all parameters by listing first the name of the distribution (exp, gamma, uniform, beta), then the parameters that define the distribution, and finally the minimum and maximum of the uniform hyper-prior. For example, `pa = “q01 exp 10 0 10”` sets an exponential distribution with mean 10 as the prior with a hyper-prior from 0 to 10 for all rates.

Continuous methods only

`directional` logical indicating whether to include a directional change parameter. Default is `FALSE`.

Continuous and ContinuousRegression methods only `lambda` number from 0 to 1 or “ML”. If a number is given, lambda will be fixed at that value. If “ML” is given, then lambda is estimated. Default is 1.

`kappa` number from 0 to 3 or “ML”. If a number is given, kappa will be fixed at that value. If “ML” is given, then kappa is estimated. Default is 1.

`delta` number from 0 to 3 or “ML”. If a number is given, delta will be fixed at that value. If “ML” is given, then delta is estimated. Default is 1.

Details

Value

An object of class MultistateML, MultistateMCMC, DiscreteML, DiscreteMCMC, ContinuousML, ContinuousMCMC, ContinuousRegressionML, or ContinuousRegressionMCMC containing information on the model and output.

Model Comparisons

Description

Compare two models with a likelihood ratio test or Bayes Factor.

Usage

```
lr.test(model1, model2)
bf.test(model1, model2)
```

Arguments

`model1`	first model to compare.
`model2`	second model to compare.

Details

Objects being compared must be of class BayesTraits ML for `lr.test` or class BayesTraitsMCMC for `bf.test`.

Value

`lr.test` returns the likelihood of each model, likelihood ratio test statistic and p-value.

`bf.test` returns the Bayes Factor and indicates the better model.

Plot Rates from Discrete Models

Description

Plot a transition-rate diagram for BayesTraits Discrete model output.

Usage

```
plot.rates(model, estimates=T, ...)
```

Arguments

`model` BayesTraits object produced by running DiscreteML or DiscreteMCMC.

`estimates` logical indicating whether to plot estimates or simply the rate labels.

`...` further arguments to be passed to plot.

Details

Objects being compared must be of class BayesTraits ML for `lr.test` or class BayesTraitsMCMC for `bf.test`.

Value

Plot depicting transition rates among character states.

Plot MCMC Output

Description

A set of four plots is made for each model parameter: 1) trace plot, 2) density plot, 3) autocorrelation plot, and 4) running mean plot. Between parameters, the user is prompted to hit return to see plots for the next parameter. The 'coda' package must be installed.

Usage

```
plot.btmcmc(model)
```

Arguments

`model` BayesTraitsMCMC object.

Details

Value

A set of four plots is displayed for each parameter. The user is prompted to hit return between each set of plots.

Kill All BayesTraits Processes

Description

Kill all BayesTraits processes running on your system from R.

Usage

```
kill.bt()
```

Arguments

Details

If MCMC output is being printed to the R console, press Esc to get your R prompt back before running `kill.bt()`

Value