Package ‘rwty’

September 4, 2014

Type Package

Title R We There Yet? A package for visualizing MCMC convergence in phylogenetics.

Version 0.1.1

Author Dan Warren <dan.l.warren@gmail.com>,Dave Swofford <david.swofford@duke.edu>,Rob Lanfear <rob.lanfear@gmail.com>

Maintainer Dan Warren <dan.l.warren@gmail.com>,Dave Swofford <david.swofford@duke.edu>,Rob Lanfear <rob.lanfear@gmail.com>

Description This package implements various tests, visualizations, and metrics for diagnosing convergence of MCMC chains in phylogenetics. It implements and automates many of the functions of the AWTY package in the R environment.

License GPL-2

Depends R (>= 3.0.1), ape, ggplot2, reshape2, phangorn, MASS, GGally

Suggests knitr,

VignetteBuilder knitr

R topics documented:

`analyze.multi` *Function for running rwty analyses on multiple chains.*

Description

This function is automatically called when analyze.rwty is called with multple chains. It runs analyze.single for each chain, and then does multi-chain analyses as well.

Usage

```
analyze.multi(chains, burnin, window.size, gens.per.tree = NA,
  treespace.points = 100, filename = NA, labels = NA, ...)
```

Arguments

`chains` A list of rwty.trees objects. `chains`

`burnin` The number of trees to eliminate as burnin `burnin`

`window.size` The length of window (in trees) for the sliding window plot `window.size`

`gens.per.tree` The number of generations per tree in the .t file.

`treespace.points`
The number of trees to plot in the treespace plot.

`filename` A name to be used for generating pdfs of output. `filename`

`labels` The name to use on plots and in generating output files. `labels`

Value

output A list of outputs from the analyze.single runs on each chain, as well as a compare.n run for all chains. Eventually we will add more multi-chain analyses.

Examples

```
data(fungus)
analyze.multi(list(run1, run2), burnin=100, window.size=20, treespace.points=100, filename="fungus.pdf", la
```

`analyze.rwty` *Analyze.rwty, the main interface for rwty analyses and plots.*

Description

This is the main user interface to rwty. It allows users to chuck in arguments for chains, burnin, window size, gens per tree, and "step", and returns an object that contains sliding window and cumulative posterior probability plots, treespace plots, and multi-chain diagnostic plots when multiple chains are provided.

Usage

```
analyze.rwty(chains, burnin, window.size, gens.per.tree = NA,
  treespace.points = 100, ...)
```

Arguments

`chains` A list of rwty.trees objects. `chains`

`burnin` The number of trees to eliminate as burnin `burnin`

`window.size` The length of window (in trees) for the sliding window plot `window.size`

`gens.per.tree` The number of generations per tree in the .t file.

`treespace.points`
The number of trees to plot in the treespace plot.

Value

output A list of outputs from the analyze.single runs on each chain, as well as a compare.n run for all chains. Eventually we will add more multi-chain analyses.

Examples

```
data(fungus)
single <- analyze.rwty(run1, burnin=100, window.size=20, treespace.points=50, filename="Run1.pdf")
multi <- analyze.rwty(list(run1, run2), burnin=100, window.size=20, treespace.points=50, labels=c("Chain1"
```

`analyze.single` *Function for running rwty analyses on single chains.*

Description

This function is automatically called when analyze.rwty is called with one chain. It runs analyze.single on the chain and returns an object with the appropriate tables and plots.

Usage

```
analyze.single(chains, burnin = 0, window.size, gens.per.tree = NA,
  treespace.points = 100, filename = NA, labels = NA, treespace = TRUE,
  ...)
```

Arguments

`chains` A single rwty.trees object. `chains`

`burnin` The number of trees to eliminate as burnin `burnin`

`window.size` The length of window (in trees) for the sliding window plot `window.size`

`gens.per.tree` The number of generations per tree in the .t file. `gens.per.tree`

`treespace.points`
The number of trees to plot in the treespace plot. `treespace.points`

`filename` A name to be used for generating pdfs of output. `filename`

`labels` The name to use on plots and in generating output files. `labels`

`treespace` Boolean to determine whether or not treespace plots are made. `treespace`

Value

output A list with tables and plots for LnL (when .p file exists), sliding window, cumulative, and treespace plots.

Examples

```
data(fungus)
analyze.single(run1, burnin=100, window.size=20, treespace.points=100, filename="fungus.pdf", labels="Chai
```

`clade.freq` *Returns clade names and frequencies*

Description

Uses ape functionality to get the frequencies and names of clades in an MCMC chain or subset thereof.

Usage

```
clade.freq(x, start, end, check.labels = TRUE)
```

Arguments

`x`	A multiPhylo or rwty.trees object
`start`	The index of the first tree to consider in calcuating frequencies
`end`	The index of the last tree to consider in calculating frequencies
`check.labels`	See documentation for ape function prop.part

Value

clade.df A data froma containing clade names and frequencies

Examples

```
data(fungus)
cfreq <- clade.freq(run1, start=10, end=100)
```

`compare.n` *Compares posterior probability estimates from multiple MCMC chains.*

Description

This function takes a set of tree files, an option array of names for those files, and a burnin argument. It returns a table with the frequencies of each clade in each of those tree files, a distance metric indicating the mean difference between clade frequencies, a plot of each chain against the others, and a translation table.

Usage

```
compare.n(x, setnames = NA, burnin)
```

Arguments

`x`	A list of rwty.trees objects.
`setnames`	A list of names for the chains.
`burnin`	The number of trees to eliminate as burnin `burnin`

Value

output A list containing a table of frequencies of each clade in each chain along with mean and sd, a distance matrix measuring consensus between chains, a translation table, and a ggpairs plot.

Examples

```
data(fungus)
compare.n(list(run1, run2), setnames=c("Chain1", "Chain2"), burnin=100)
```

`continuous.distance` *Various small utility functions*

Description

A bunch of tiny utility functions that we need off and on. Not intended for user interaction.

Usage

```
continuous.distance(two.trees)
```

`cumulative.freq` *Cumulative frequencies of clades in an MCMC chain*

Description

Calculates the posterior probability estimate of each clade in an MCMC chain as a function of chain length.

Usage

```
cumulative.freq(tree.list, burnin = 0, window.size, gens.per.tree = 1,
  slide.freq.table = NULL, ...)
```

Arguments

`tree.list`	A single rwty.trees or multiPhylo object. `chains`
`burnin`	The number of trees to eliminate as burnin `burnin`
`window.size`	The length of window (in trees) for the sliding window plot `window.size`
`gens.per.tree`	The number of generations per tree in the .t file.
`slide.freq.table`	A table from slide.freq. If a table is not provided, it will be built.

Value

output A list containing a table of cumulative frequencies and a clade translation table

Examples

```
data(fungus)
cumulative.data <- cumulative.freq(run1$trees, burnin=100, window.size=20, gens.per.tree=10000)
```

`load.trees` *Custom functions to load tree lists so that rwty can do basic processing on the way in.*

Description

Loads trees, looks for a .p file of tree likelihoods, returns and rwty.trees object containing both

Usage

```
load.trees(file, type = "nexus", gens.per.tree = NA, trim = 1,
  skiplines.p = 1)
```

Arguments

`file` A path to a .t file containing an MCMC chain of trees

`type` An argument that designates the type of tree file. If "nexus", trees are loaded using ape's read.nexus function. Otherwise, it's read.tree.

`gens.per.tree` The number of generations separating trees. If not provided, RWTY will attempt to calculate it automatically.

`trim` Used for thinning the chain. If a number N is provided, RWTY keeps every Nth tree.

`skiplines.p` The number of lines that must be skipped to get to the header of the p file. MrBayes, for instance, prints a comment line at the top of the p file, so MrBayes files should be read in with a skiplines.p value of 1.

Value

output An rwty.trees object containing the multiPhylo and the table of values from the .p file if available.

Examples

```
#load.trees(file="mytrees.nex", type="nexus")
```

parse.clades *Rename clades for easy recall*

Description

Converts a list of clades (e.g., "1 2 3 4" as a clade) and returns a list of parsed clades, converting numbers to names using a set of trees. Called internally by the slide and cumulative analyses, not user-facing.

Usage

```
parse.clades(clades, treelist)
```

Arguments

`clades`	A list of clades, as in the first column of a cladetable in an rwty.slide or rwty.cumulative object
`treelist`	A list of trees, used for getting tip names

Value

output A list of clades with parsed tip names

plot.cladeprobs *New style plotting of cumulative and slide objects*

Description

Takes a table from cumulative.freq or slide.freq as input. Numclades gives the number of clades to plot, starting from the top. Since cumulative.freq and slide.freq both sort by sd these will by default be the most variable clades.

Usage

```
## S3 method for class cladeprobs
plot(input.table, numclades = 20)
```

Arguments

`x`	An rwty.slide or rwty.cumulative object
`numclades`	The number of clades to plot. The clades with the highest sd in clade frequency are plotted first, so numclades = 10 will be the 10 most variable clades in the chain.

Value

thisplot Returns a ggplot object.

Examples

```
data(fungus)
slide.data <- slide.freq(run1$trees, burnin=100, window.size=20, gens.per.tree=10000)
cpplot <- plot.cladeprobs(input.table = slide.data$slide.table, numclades=25)
```

plot.cladevar *Plots distribution of variance in posterior probability estimates across clades.*

Description

Plots the distribution of variances in posterior probability estimates across clades in a chain as a function of chain length.

Usage

```
## S3 method for class cladevar
plot(input.table, numclades = nrow(input.table))
```

Arguments

`input.table`	Either a rwty.cumulative or rwty.slide object, or a cumulative or slide table from one of those objects.
`numclades`	The number of clades to calculate stats for. Defaults to all clades in the table.

Value

thisplot A ggplot object

Examples

```
data(fungus)
slide.data <- slide.freq(run1$trees, burnin=100, window.size=20, gens.per.tree=10000)
cvplot <- plot.cladevar(slide.data$slide.table, numclades=100)
```

plot.treespace *Plot chains in treespace.*

Description

This function will take an mds.treespace object and produce plots of chains in treespace.

Usage

```
## S3 method for class treespace
plot(points)
```

Arguments

`points`	A set of points to plot `points`

Value

p A ggplot object

`slide.freq` *Sliding window measurements of clade posterior probabilities.*

Description

This function takes sliding windows of a specified length over an MCMC chain and calculates the posterior probability of clades within that window. It allows users to see whether the chain is visiting different areas of treespace.

Usage

```
slide.freq(tree.list, burnin = 0, window.size, gens.per.tree = 1, ...)
```

Arguments

`tree.list`	A rwty trees object or multiPhylo `tree.list`
`burnin`	The number of trees to eliminate as burnin `burnin`
`window.size`	The number of trees to include in each window `window.size`
`gens.per.tree`	The number of steps in the MCMC chain corresponding to a tree in the tree list. `gens.per.tree`

Value

rwty.slide An object containing the frequencies of clades in the sliding windows, a translation table that converts clade groupings to factors, and a plot.

Examples

```
data(fungus)
slide.data <- slide.freq(run1$trees, burnin=100, window.size=20, gens.per.tree=10000)
```

`tree.dist.matrix` *Tree distance matrix calculation*

Description

This function takes a list of trees and returns a distance matrix populated with Robinsoun-Foulds tree distances between all trees in the list.

Usage

```
tree.dist.matrix(trees, treenames = names(trees))
```

Arguments

`trees`	a multiPhylo object `trees`
`treenames`	A vector of names for the trees `treenames`

Value

RF A distance matrix of RF distances

Examples

```
data(fungus)
tree.dist.matrix(list(run1$trees[[1]], run1$trees[[2]], run1$trees[[3]]), treenames=c("tree1", "tree2", "tr
```

tree.distance *Returns distance between two trees*

Description

Takes a list of two trees, returns Robinson-Foulds distance between them.

Usage

```
tree.distance(two.trees)
```

Arguments

`two.trees`	A list containing two trees `two.trees`

Value

d A distance measurement

Examples

```
data(fungus)
tree.distance(list(run1$trees[[1]], run1$trees[[2]]))
```

treespace.single *MDS scaling of treespace for a single tree list.*

Description

This function constructs a distance matrix from a list of trees and uses multi-dimensional scaling to collapse it to a two- dimensional tree space for plotting.

Usage

```
treespace.single(trees, gens, ptable = NULL)
```

Arguments

`trees`	A multiphylo object `trees`
`gens`	A list of the generation represented by each tree, for use in plotting
`ptable`	A table of p values to use for plotting likelihoods `p.file`

Value

Returns a list containing the mds table and a plot.

Examples

```
data(fungus)
burnin <- 100
#We have to start by trimming the chain and p table down to a reasonable number of trees
mdstrees <- run1$trees[seq((burnin + 1), length(run1$trees), by = 20)]
#Now were going to get a list of generations matching the trees retained from the chain
gens <- as.numeric(unlist(regmatches(names(run1$trees), gregexpr(\\(?[0-9]+, names(run1$trees)))))
gens <- gens[seq((burnin + 1), length(run1$trees), by = 20)]
#Finally were going to cut down the p table to just the bits were going to use
mdsptable <- run1$ptable[seq((burnin + 1), length(run1$trees), by = 20),]
this.treespace <- treespace.single(mdstrees, gens=gens, ptable=mdsptable)
```

Index