Package 'Rsampletrees'

March 2, 2020

Type Package

Title MCMC Sampling of Gene Genealogies Conditional on Genetic Data

Version 1.0.3

Date 2020-02-21

Author Kelly Burkett, Brad McNeney, Jinko Graham, Jean-Francois Lefebvre

Maintainer Kelly Burkett <Rsampletrees@gmail.com>

Description Sample ancestral trees conditional on phased or unphased SNP genotype data. The actual tree sampling is done using a C++ program that is launched within R. The package also contains functions for specifying the tree-sampling settings (pre-processing) and for storing and manipulating the sampled trees (post-processing). More information about 'sampletrees' can be found at <http://stat.sfu.ca/statgen/research/sampletrees.html>.

License GPL (>= 2)

Imports Rcpp (>= 0.12.9), haplo.stats, ape

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-03-02 20:30:02 UTC

R topics documented:

`Rsampletrees-package` *MCMC sampling of gene genealogies conditional on genetic data*

Description

Sample ancestral trees conditional on phased or unphased SNP genotype data. The actual tree sampling is done using a C++ program that is launched within R. The package also contains functions for specifying the tree-sampling settings (pre-processing) and for storing and manipulating the sampled trees (post-processing). More information about sampletrees can be found at http://stat.sfu.ca/statgen/research/sampletrees.html.

Details

Package:	Rsampletrees
Type:	Package
Version:	1.0.3
Date:	2020-02-21
License:	GPL-2

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016.

`addTrees` *Read in trees and add them to the tree output object*

Description

This function will read in the trees from a from the file specified in the treeoutput object and store them in the treeoutput object. The user can specify that all or a subset of the trees be read in, as described in the Details section.

Usage

```
addTrees(output, all=TRUE, lines=NULL, start=1, end=NULL, nlines=NULL)
```

Arguments

output	An object of class 'treeoutput'
all	If TRUE (default), all trees in the file will be read in. If FALSE, the trees specified by lines or start/stop/nlines will be read in
lines	A vector containing the line numbers of the tree file to be read in
start	The first line of the tree file to read in
end	The last line of the tree file to read in
nlines	The number of lines to be read in

Details

The trees are read in using the Rsampletrees readTree() function, which in turn calls the read.tree() function from the ape package.

To read in all of the trees in the file, use the all=TRUE option. If all=FALSE, then a subset of trees are read in using either the lines, start/stop or start/nlines options.

For the lines option, the 'lines' argument will consist of a vector of positive integers. These numbers correspond to the lines in the tree file rather than to the MCMC sample number. For example, say that a chain of length 2000 is run, with a thinning interval of 100 and no burn-in. The lines of the tree file will be the 100th, 200th, 300th, etc. trees. To read in the first 4 trees, set lines=1:4 and not lines=c(100,200,300,400). This option is useful if non-consecutive rows are to be read in.

If the start/stop/nlines options are used, 'start' should be set to the first row number to be read in. If a stop line is provided, then all lines between and including 'start' and 'stop' will be read in. If 'nlines' is provided, than a total of 'nlines' will be read in, starting from the row given by 'start'.

Value

An object of class 'treeoutput' with the trees stored in the rawdata component

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

readOutput, readTrees

Examples

```
#\dontrun{
# Read in the settings; Must change the RunName so that the example files can be found

#system.file("Examples/example_g_pars",package="Rsampletrees")
filename=paste(path.package("Rsampletrees"),"/extdata/example_g_pars",sep="")
runpars=readArgs(filename, check=FALSE)
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName,sep="/")
runpars=changeArgs(runpars, RunName=runname)

# 1. Read in all the trees; may be slow
results=readOutput(argobj=runpars)
results=addTrees(results)
length(results$rawdata$Trees)

# 2. Read in a selection of lines
results=readOutput(argobj=runpars)
results=addTrees(results, all=FALSE, lines=c(5,10,31))
length(results$rawdata$Trees)

# 3. Read in trees from lines 5 to 15
results=readOutput(argobj=runpars)
results=addTrees(results, all=FALSE, start=5, end=15)
length(results$rawdata$Trees)

# 4. Read in 20 trees, starting at line 10
results=readOutput(argobj=runpars)
results=addTrees(results, all=FALSE, start=10, nlines=20)
length(results$rawdata$Trees)
#}
```

`addTreeStat` *Add tree statistics to a treeoutput object*

Description

This function computes a summary statistic on each of the trees given in the treeoutput object. The results are stored in the procdata component of the treeoutput object.

Usage

```
addTreeStat(output, myfunc, funcname=NULL, maxlines=1000, treerange=NULL, ...)
```

Arguments

`output`	An object of class 'treeoutput'
`myfunc`	The function to be applied to each tree

funcname	The name of the function to be used for the column name of the TreeStat data frame
maxlines	The maximum number of lines of the tree file to read in at a time. This avoids large tree files from fully being read in.
treerange	A vector giving the indices of the trees to which the function should be applied. Only used if the trees haven't been read in
...	further arguments that can be passed to treeapply.

Details

As with the function treeapply, the tree statistics can be computed on a subset of all the trees (provided by treerange). Note, however, that if TreeStats already exists due to previous call to addTreeStat(), the function is applied to the same trees as indexed in the first column of TreeStats even if these are different from treerange (a warning is given). This ensures a data frame of the correct dimensions.

Value

Returns an object of class 'treeoutput' that is the same as output, except that the tree statistics have been added to the data frame TreeStats in the procdata component.

The data frame TreeStats consists of a column for the tree index and a column for each of the tree statistics computed.

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

readOutput, treeapply

Examples

```
# A function that computes the time to MRCA of a tree using the ape package
require(ape)
mrca.age=function(tree)
{
return(coalescent.intervals(tree)$total.depth)
}
#\dontrun{
 # Read in the settings; Must change the RunName so that the example files can be found
#system.file("Examples/example_g_pars",package="Rsampletrees")
 filename=paste(path.package("Rsampletrees"),"/extdata/example_g_pars",sep="")
 runpars=readArgs(filename, check=FALSE)
```

```
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
 runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName,sep="/")
 runpars=changeArgs(runpars, RunName=runname)

 # Read in the output and add the summary statistic
 results=readOutput(argobj=runpars)
 results=addTreeStat(results, myfunc=mrca.age, funcname="Time.to.MRCA")
#}
```

`changeArgs.treeoutput` *Modify the settings stored in a sampletrees treeoutput object*

Description

Modify the settings stored in a sampletrees treeoutput object.

Usage

```
## S3 method for class 'treeoutput'
changeArgs(object,...)
```

Arguments

`object`	An object of class 'treeoutput'
`...`	Optional tag names and the value they should be set to. See Details section.

Details

The function changeArgs modifies the settings values from a run of sampletrees (the settings are stored in the runinfo component of the treeoutput object). The format for passing arguments to this function is: TagName=Value. The names are given in the Value section.

Value

Returns an object of class 'treeoutput' with the runinfo settings modified as specified.

The settings names in the runinfo component are:

`RunName`	Run name for the sampletrees run (Default="Run")
`Seed`	Initial seed for sampletrees run. (Default=NA)
`DataType`	The type of the data file g=genotype h=haplotype. (Default="h")
`DataFile`	The name of the file containing the haplotype or genotype data. DEFAULT=NA but the user must change this value before running sampletrees()
`LocationFile`	The name of the file containing the genomic locations (in base pairs) of the SNP markers. Default=NA but the user must change this value before running sampletrees()

`WeightFile` The name of the file containing the probabilities for sampling each of the 7 updates. Default=NA but the user must change this value before running sampletrees(). See setWeights for more information.

`FocalPoint` The location of the focal point. Default=NA but the user must change this value before running sampletrees()

`ChainLength` How long to run the chain. (Default=1000)

`BurnIn` Discard the first 'BurnIn' samples. (Default=100)

`Thinning` Return output every 'Thinning'th sample. (Default=1)

`InitialTheta` Initial value for mutation rate theta. (Default=1)

`MinTheta` Minimum for Uniform prior for theta. (Default=0.0001)

`MaxTheta` Maximum for Univorm prior for theta. (Default=10)

`InitialRho` Initial value for recombination rate rho. (Default=0.0004)

`ScaleRho` Scale parameter for gamma prior for rho. (Default=0.1)

`ShapeRho` Shape parameter for gamma prior for rho. (Default=1)

`InitialTreeFile`
Name for a file containing initial tree data for a run of sampletrees. These are typically available from a previous run of sampletrees. If DataType="g", initial haplotype configurations will be taken from this file rather than one specified by InitialHaploFile. (Default=NA)

`RandomTree` Indicates whether initial tree generated by randomly connecting nodes. (Default=FALSE)

`HaploFreqFile` The name of the file with the haplo frequency estimates to be used only if DataType="g". See estimateHapFreqs() for more information. Default=NA but user must change this value if DataType is 'g'.

`InitialHaploFile`
Name for an optional file containing the initial haplotype configurations for the sampletrees run (optional if DataType="g"). Each row of this file corresponds to a haplotype, there are 2 rows/individual and rows must be in the same order as in DataFile. (Default=NA)

`HaploListFile` Name for a optional file containing a list of likely haplotypes (optional if DataType="g"). Each row corresponds to a haplotype. (Default=NA)

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

changeArgs.pars, checkArgs

Examples

```
#\dontrun{
 #system.file("Examples/example_g_pars",package="Rsampletrees")
filename=paste(path.package("Rsampletrees"),"/extdata/example_g_pars",sep="")
runpars=readArgs(filename, check=FALSE)
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName, sep="/")
runpars=changeArgs(runpars, RunName=runname)
results=readOutput(argobj=runpars)
results=changeArgs(results, RunName="NewName")
#}
```

---

checkArgs *Error checking for the arguments to sampletrees*

Description

This function is used to check the arguments to be used for a run of sampletrees. If the options are properly specified, then sampletrees can be expected to run without errors from improper input files.

Usage

```
checkArgs(args)
```

Arguments

args	An object of class 'pars' with the arguments for the sampletrees run

Details

Extensive checking is done, including:

- Non-default values have been provided for settings requiring user-specified values

- All input files exist

- All numeric settings are set to numeric values

- The format of the genotype and/or sequence files are correct

- The file with SNP locations contains only numeric values in increasing order, and that the number of locations matches the number of SNPs in the genotype/sequence file

- Focal point is set to a location in the region containing the SNPs

- Weights sum to 1 and the proposal indices are in the right range

Value

Returns an object of class 'pars' with the 'clean' variable set to TRUE if the arguments in 'args' pass the error check and FALSE otherwise.

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

Examples

```
# Will produce error messages
runpars=newArgs()
runpars=checkArgs(runpars)

# Will not produce errors
#system.file("Examples/sequences_Theta8_Rho8.txt",package="Rsampletrees")
datname=paste(path.package("Rsampletrees"),"/extdata/sequences_Theta8_Rho8.txt",sep="")
#system.file("Examples/locations_Theta8_Rho8.txt",package="Rsampletrees")
locname=paste(path.package("Rsampletrees"),"/extdata/locations_Theta8_Rho8.txt",sep="")
#system.file("Examples/weights-h.txt", package="Rsampletrees")
weightname=paste(path.package("Rsampletrees"),"/extdata/weights-h.txt",sep="")
runpars=newArgs(DataFile=datname, DataType="h", WeightFile=weightname,
LocationFile=locname,RunName="Test-h",FocalPoint=10000)
runpars=checkArgs(runpars)
```

---

estimateHap *Estimate and write haplotype probabilities and initial values to files*

Description

This function is only used when the data type is 'g' (genotype). Estimate the two-locus haplotype probabilities from the genotype data. Optionally set initial haplotype configurations and a list of likely haplotypes for the run of sampletrees. This function requires the R package "haplo.stats" for estimating the haplotype frequencies from the genotype data.

Usage

```
estimateHap(args, HaploFreqFile, InitialHaplos = TRUE,
InitialHaploFile = "initialhaps", HaploList = TRUE,
HaploListFile = "initialhaplist", tol = 1e-05)
```

Arguments

args	An object of class 'pars' with the sampletrees settings
HaploFreqFile	The name of the output file for the two-locus haplotype probability estimates

`InitialHaplos` If TRUE, sample the initial haplotype configuration using the posterior probabilities of each configuration for each individual estimated using haplo.em() (Default=TRUE)

`InitialHaploFile` File name for the initial haplotype configurations

`HaploList` If TRUE, create a list of likely haplotypes for a run of sampletrees with genotype data (Default=TRUE). This list will include haplotypes estimated to have a probability greater than 'tol'

`HaploListFile` File name for the haplotype list

`tol` Haplotypes with estimated probability greater than this value will be included in the list of likely haplotypes

Details

This function is only used when the data type is genotype ('g').

The two-locus haplotype probabilities are estimated using the haplo.em() function in the haplo.stats package. This package uses an EM approach that has been adapted to handle estimation of haplotype probabilities when the haplotypes are made up of many loci. The haplotype probabilities are estimated for haplotypes containing all loci. The probability for a given two-locus haplotype is then computed by summing up probabilities for the full haplotypes having the given two-locus haplotype (possible haplotypes are 00, 01, 10 or 11). These probabilities are computed for all adjacent pairs of loci.

When using genotype data, it is recommended that a list of likely or known haplotypes and an initial configuration be provided to sampletrees in order to improve MCMC mixing. These can optionally be initialized using the output from haplo.em() if HaploList and InitialHaplos are set to TRUE. The list of likely haplotypes will contain those haplotypes that have probability estimated to be above a threshold (set by 'tol'). The initial haplotype configuration for all individuals is initialized by sampling a configuration based on the estimated posterior probabilities of each haplotype configuration for each individual.

Value

This function writes the estimated haplotype data to the specified files and returns

`args` An object of class 'pars' with the haplotype options set to those specified by the call to this function

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

Examples

```
## Not run:
datname=paste(path.package("Rsampletrees"),"/extdata/geno_Theta8_Rho8.txt",sep="")
locname=paste(path.package("Rsampletrees"),"/extdata/locations_Theta8_Rho8.txt",sep="")
weightname=paste(path.package("Rsampletrees"),"/extdata/weights-g.txt",sep="")

runpars=newArgs(DataFile=datname, DataType="g", LocationFile=locname, WeightFile=weightname,
    RunName="Test-g",FocalPoint=10000)

# This will create the three files"EM-hapfreqs", "EM-initial.dat", "EM-known_haplotypes"
runpars=estimateHap(runpars,"EM-hapfreqs",InitialHaploFile="EM-initial.dat",
HaploListFile="EM-known_haplotypes")


## End(Not run)
```

---

launch.sampletrees *Launches the c++ sampletrees program using arguments previously given and checked.*

Description

This function performs the MCMC tree sampling by launching the c++ sampletrees program. Linking of the pre_sampletrees functions to the c++ sampletrees program is done using Rcpp package. For more information about the sampletrees program, please see the package vignette.

Usage

```
launch.sampletrees(args, addtrees=FALSE)
```

Arguments

args	An object of class 'pars' with the arguments for the sampletrees run
addtrees	If TRUE, store the trees in the output object (default=FALSE)

Details

Sampletrees is a computationally intensive program. As such, it will write important MCMC output to output files as a back-up. Upon completion of the tree sampling, launch.sampletrees will read the output into R as a treeoutput object.

Value

An object of class 'treeoutput', which is a list made up of three components:

1) runinfo - a copy of argobj

2) rawdata

3) procdata

The component 'rawdata' consists of

i	Iteration numbers of the MCMC samples
Theta	A vector of sampled theta values (mutation rate)
Rho	A vector of sampled rho values (recombination rate)
Trees	Either a string containing the name of the tree file or a list of class 'multiPhylo' containing the trees (if addtrees=TRUE). See the ape package documentation for more information on the 'multiPhylo' class.

The component 'procdata' (processed data) is also a list, initially made up only of a matrix with the acceptance proportions for each update type. Tree statistics may be added to procdata by addTreeStat

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

Examples

```
## Not run:
oldPath=getwd()
setwd(paste(path.package("Rsampletrees"),"/extdata/",sep=""))
runpars=readArgs("example_h_pars")
runtree=launch.sampletrees(runpars)
setwd(oldPath)

## End(Not run)
```

---

merge.treeoutput *Merge the results from two runs of sampletrees*

Description

This function can be used to concatenate the results of two consecutive runs of sampletrees.

Usage

```
## S3 method for class 'treeoutput'
merge(x, y, runname=NULL, ...)
```

Arguments

x	An object of class 'treeoutput' with results from the first sampletrees run
y	An object of class 'treeoutput' with results from the second sampletrees run.
runname	The name to give the merged run
...	additional arguments to merge; currently unused

Details

The function restartRun can be used to set options for starting sampletrees where a previous run finished. Once the second run is complete, it is useful to merge the processed output of the two runs. This function does not merge the actual files; instead, a new object of class 'treeoutput' is created that stores the run information, the merged Theta values, Rho values, and any tree summaries that were computed. Note that it assumes that the same tree summaries were computed on both of the runs. To save the merged results use writeTreeoutput.

In order to set values for the first sample and last sample, the value of the thinning interval is used. The computation assumes that the thinning interval is the same for both sets of output. A warning is given if they are not the same, but computation is not stopped; the thinning for the first set of output is assumed.

Value

Returns a list of class 'treeoutput' with values set to those in output1 and output2.

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

restartRun, writeTreeoutput

Examples

```
#\dontrun{
#system.file("Examples/example_h_pars",package="Rsampletrees")
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
filename=paste(path.package("Rsampletrees"),"/extdata/example_h_pars",sep="")
runpars=readArgs(filename, check=FALSE)
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName, sep="/")
runpars=changeArgs(runpars, RunName=runname)
results1=readOutput(argobj=runpars)

#system.file("Examples/example_h_2_pars",package="Rsampletrees")
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
filename=paste(path.package("Rsampletrees"),"/extdata/example_h_2_pars",sep="")
runpars=readArgs(filename, check=FALSE)
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName, sep="/")
runpars=changeArgs(runpars, RunName=runname)
results2=readOutput(argobj=runpars)
newresults=merge(x=results1, y=results2, runname="Merge2000")

#}
```

newArgs *Create or modify a sampletrees settings object*

Description

Create or modify a settings object containing the arguments for a run of the C++ program sampletrees.

Usage

```
newArgs(...)

## S3 method for class 'pars'
changeArgs(object,...)
```

Arguments

object	An object of class 'pars'
...	Tags and the values they should be set to. A list of the tag names is given in the Value section

Details

The function newArgs() initializes a list object of class 'pars' that contains the settings for a run of sampletrees. The user can optionally pass arguments to newArgs to change some of the settings from their default values. The format for passing arguments to newArgs is: TagName=Value. The allowable tag names and their default values for an object of type 'pars' are given below in the Value section.

The function changeArgs modifies the values in a settings object of class 'pars'. The arguments are passed to changeArgs in same way as to newArgs.

Note that although both functions allow list elements to be set to default values, before running sampletrees the user will have to provide non-default values for some elements (DataFile, LocationFile, WeightFile, FocalPoint). Using non-default values is recommended for ChainLength/BurnIn/Thinning as the default values are set to ensure short test runs.

Value

Returns an object of class 'pars' which is a list with elements:

RunName	Run name for the sampletrees run (Default="Run")
Seed	Initial seed for sampletrees run. (Default=NA)
DataType	The type of the data file g=genotype h=haplotype. (Default="h")
DataFile	The name of the file containing the haplotype or genotype data. DEFAULT=NA but the user must change this value before running sampletrees

`LocationFile` The name of the file containing the genomic locations (in base pairs) of the SNP markers. Default=NA but the user must change this value before running sampletrees

`WeightFile` The name of the file containing the probabilities for sampling each of the 7 updates. Default=NA but the user must change this value before running sampletrees. See setWeights for more information.

`FocalPoint` The location of the focal point. Default=NA but the user must change this value before running sampletrees

`ChainLength` How long to run the chain. (Default=1000)

`BurnIn` Discard the first 'BurnIn' samples. (Default=100)

`Thinning` Return output every 'Thinning'th sample. (Default=1)

`InitialTheta` Initial value for mutation rate theta. (Default=1)

`MinTheta` Minimum for Uniform prior for theta. (Default=0.0001)

`MaxTheta` Maximum for Univorm prior for theta. (Default=10)

`InitialRho` Initial value for recombination rate rho. (Default=0.0004)

`ScaleRho` Scale parameter for gamma prior for rho. (Default=0.1)

`ShapeRho` Shape parameter for gamma prior for rho. (Default=1)

`InitialTreeFile`
Name for a file containing initial tree data for a run of sampletrees. These are typically available from a previous run of sampletrees. If DataType="g", initial haplotype configurations will be taken from this file rather than one specified by InitialHaploFile. (Default=NA)

`RandomTree` Indicates whether initial tree generated by randomly connecting nodes. (Default=FALSE)

`HaploFreqFile` The name of the file with the haplo frequency estimates to be used only if DataType="g". See estimateHapFreqs() for more information. Default=NA but user must change this value if DataType is 'g'.

`InitialHaploFile`
Name for an optional file containing the initial haplotype configurations for the sampletrees run (optional if DataType="g"). Each row of this file corresponds to a haplotype, there are 2 rows/individual and rows must be in the same order as in DataFile. (Default=NA)

`HaploListFile` Name for a optional file containing a list of likely haplotypes (optional if DataType="g"). Each row corresponds to a haplotype. (Default=NA)

`clean` An indicator for whether the list passes the checks for a clean sampletrees run. In order to ensure no errors with sampletrees, the user should not modify this value and instead should run checkPars(). (Default=FALSE)

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

checkArgs

Examples

```
runpars=newArgs(DataFile="sequences_Theta8_Rho8.txt", DataType="h",
LocationFile="locations_Theta8_Rho8.txt",RunName="Test-h",FocalPoint=10000)
runpars=changeArgs(runpars, Seed=1938474, WeightFile="weights-h.txt")
```

---

plot.treeoutput *Plot selected output of a sampletrees run*

Description

This function produces the following plots:

1) Barchart giving the acceptance proportions for each update type

2) Traceplot of mutation rate (theta)

3) Traceplot of recombination rate (rho)

If tree statistics have been computed on the sampled trees, traceplots of the tree statistics will also be produced.

Usage

```
## S3 method for class 'treeoutput'
plot(x, oneperpage = FALSE, asktoplot = FALSE, layoutmat = NULL,
statnames = NULL, ...)
```

Arguments

x	An object of class 'treeoutput' containing the settings and output from a sampletrees run
oneperpage	Only one plot per page (Optional)
asktoplot	Plots are printed to screen after hitting enter (Optional)
layoutmat	A matrix giving the layout for the plots on the page. This is passed to the layout() function (Optional)
statnames	Names for the tree statistics for plotting.
...	Other arguments to be passed to plot.

Details

The acceptance proportions for update types 1,2, and 4-7 are the number of accepted changes divided by the number of times a given update was performed. Update type 3 proposes changes to each node of the tree in turn. The acceptance proportion for update type 3 is therefore the average of the proportion of nodes that have an update accepted.

Value

The three or more plots are plotted in the graphics window.

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

layout

Examples

```
#\dontrun{
require(ape)
mrca.age=function(tree)
{
return(coalescent.intervals(tree)$total.depth)
}

#system.file("Examples/example_g_pars", package="Rsampletrees")
filename=paste(path.package("Rsampletrees"),"/extdata/example_g_pars",sep="")
runpars=readArgs(filename, check=FALSE)

# Must change the path so that the output can be found
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName,sep="/")
runpars=changeArgs(runpars, RunName=runname)

# Read in the output
results=readOutput(argob=runpars)

# Plot the output
plot(results)
plot(results, layoutmat=matrix(c(1,1,2,3),byrow=TRUE,nrow=2))
plot(results, oneperpage=TRUE, asktoplot=TRUE)

# Add a tree statistic
results=addTreeStat(results, myfunc=mrca.age, funcname="Time.to.MRCA")
plot(results)

#}
```

print.pars *Nice display of tags and values of a sampletrees settings object*

Description

This function prints out the values in a sampletrees settings object in a tabular format. The first column of the table is the names of all the tags. The second column gives their corresponding values.

Usage

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

Arguments

x	An object of class 'pars'
...	additional arguments to print; currently unused

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

summary.pars, checkArgs, print.treeoutput

Examples

```
runpars=newArgs()
print(runpars)
runpars
```

print.treeoutput *Nice display of a treeoutput object*

Description

This function prints out information about a treeoutput object. The component runinfo is an object of class 'pars' and is printed using print.pars. The total number of samples and their first/last index is also printed.

Usage

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

Arguments

x	An object of class 'treeoutput'
...	additional arguments to print; currently unused

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

print.pars

Examples

```
#\dontrun{
#system.file("Examples/example_g_pars",package="Rsampletrees")
filename=paste(path.package("Rsampletrees"),"/extdata/example_g_pars",sep="")

runpars=readArgs(filename, check=FALSE)
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName,sep="/")
runpars=changeArgs(runpars, RunName=runname)
results=readOutput(argobj=runpars)

print(results)

#}
```

readArgs *Read arguments for a sampletrees run from a settings file*

Description

This function is used to read in settings for a sampletrees run that have been previously saved to a file.

Usage

```
readArgs(filename, check=TRUE)
```

Arguments

filename	The name of the input file with the sampletrees settings
check	If TRUE, error checking of the settings will be done (default=TRUE)

Value

Returns an object of class 'pars' with settings from the input file

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

Examples

```
filename=paste(path.package("Rsampletrees"),"/extdata/example_g_pars",sep="")
runpars=readArgs(filename, check=FALSE)
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName, sep="/")
runpars=changeArgs(runpars, RunName=runname)
print.pars(runpars)
```

readOutput *Read in sampletrees output*

Description

This function is used to read in results from a run of sampletrees. Either the object with the settings for the run or the file containing these settings is passed to the function. A new object is created that contains a component that stores the run settings, plus additional components storing the output, the sampled values for theta and rho, and a matrix with acceptance proportions. Note that the trees themselves are by default not read in as these files can be very large.

Usage

```
readOutput(treeobj=NULL, argobj=NULL, argfile=NULL, addtrees=FALSE)
```

Arguments

treeobj	A 'treeoutput' object from a previous sampletrees runs
argobj	A settings object of class 'pars' describing the sampletrees run
argfile	A file containing the settings describing the sampletrees run
addtrees	If TRUE, store the trees in the output object (default=FALSE)

Details

One of either argobj or argfile must not be NULL.

Value

An object of class 'treeoutput', which is a list made up of three components:

1) runinfo - a copy of argobj

2) rawdata

3) procdata

The component 'rawdata' consists of

i	Iteration numbers of the MCMC samples
Theta	A vector of sampled theta values (mutation rate)
Rho	A vector of sampled rho values (recombination rate)
Trees	Either a string containing the name of the tree file or a list of class 'multiPhylo' containing the trees (if addtrees=TRUE). See the ape package documentation for more information on the 'multiPhylo' class.

The component 'procdata' (processed data) is also a list, initially made up only of a matrix with the acceptance proportions for each update type. Tree statistics may be added to procdata by addTreeStat

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

addTreeStat, addTrees

Examples

```
#\dontrun{
#system.file("Examples/example_g_pars", package="Rsampletrees")
filename=paste(path.package("Rsampletrees"),"/extdata/example_g_pars",sep="")
runpars=readArgs(filename, check=FALSE)

# Must change the path so that the output can be found
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName, sep="/")
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
runpars=changeArgs(runpars, RunName=runname)

# Read in the output
results=readOutput(argobj=runpars)

#}
```

---

readTrees *Read in trees from a run of sampletrees.*

Description

This function is used to read in trees from a sampletrees. Trees are stored as a list of class 'multiPhylo'. The user can specify that all or a subset of the trees be read in; see Details, below.

Usage

```
readTrees(output=NULL, filenames=NULL,  all=TRUE, lines=NULL,
start=1, end=NULL, nlines=NULL)
```

Arguments

output	A treeoutput object
filenames	A vector of names for the tree files
all	If TRUE, all trees in the file(s) will be read in. If FALSE, the trees specified by lines or start/stop/nlines will be read in

lines	A vector containing the line numbers of the tree file to be read in
start	The first line of the tree file to read in
end	The last line of the tree file to read in
nlines	The number of lines to be read in

Details

The trees are read in using the read.tree() function from the ape package, which stores results as an object of class 'multiPhylo'.

The user has the option to provide either a vector of tree names or a treeoutput object for reading in the trees. At least one of filenames or output must not be NULL.

To read in all of the trees in the file, use the all=TRUE option. If all=FALSE, then a subset of trees are read in using either the lines, start/stop or start/nlines options.

For the lines option, the 'lines' argument will consist of a vector of positive integers. These numbers correspond to the lines in the tree file rather than to the MCMC sample number. For example, say that a chain of length 2000 is run, with a thinning interval of 100 and no burn-in. The lines of the tree file will be the 100th, 200th, 300th, etc. trees. To read in the first 4 trees, set lines=1:4 and not lines=c(100,200,300,400). This option is useful if non-consecutive rows are to be read in.

If the start/stop/nlines options are used, 'start' should be set to the first row number to be read in. If a stop line is provided, then all lines between and including 'start' and 'stop' will be read in. If 'nlines' is provided, than a total of 'nlines' will be read in, starting from the row given by 'start'.

Value

An object of class 'multiPhylo' which is a list of tree elements.

Note

Note that this function does not store the trees in a treeoutput object; use addTrees instead.

Author(s)

Kelly Burkett; uses functions from APE package, maintained by Emmanuel Paradis

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

See Also

read.tree, addTrees

Examples

```
#\dontrun{
treesname=paste(path.package("Rsampletrees"),"/extdata/Example-h_trees.out",sep="")
#system.file("Output/Example-h_trees.out", package="Rsampletrees")

# Read in all the trees in the file; may be slow
mytrees=readTrees(filename=treesname)
length(mytrees)

# Read in the first and 90th line
mytrees=readTrees(filename=treesname,all=FALSE, lines=c(1,40))
names(mytrees)

# Read in lines  2-4
mytrees=readTrees(filename=treesname,all=FALSE, lines=2:4)
names(mytrees)

# Read in three 3 starting at line 2
mytrees=readTrees(filename=treesname, all=FALSE, start=2, nlines=3)
names(mytrees)

# Read in lines 2-4
mytrees=readTrees(filename=treesname, all=FALSE, start=2, end=4)
names(mytrees)

#}
```

---

restartRun *Set up a settings object for continuing a previous sampletrees run*

Description

This function is used to initialize settings in order to restart a sampletrees run. The initial values for theta, rho and the initial tree are taken from the final sampled values of a previous sampletrees run.

Usage

```
restartRun(newrunname, oldargs = NULL, argfile = NULL, extrait = NULL,
totalsamples = NULL)
```

Arguments

newrunname	The name to associate with the new run. Output files will have this run name as prefix
oldargs	An object of class 'pars' with the settings for the previous sampletrees run
argfile	The name of the settings file used for the previous sampletrees run
extrait	The number of additional iterations desired
totalsamples	The total number of iterations desired in the previous and new run

Details

The settings of the previous sampletrees runs can be specified in terms of the settings object or a file name. Therefore, at least one of 'oldargs' or 'argfile' must not be NULL.

The desired number of MCMC samples can be specified either in terms of the additional iterations to run ('extrait') or in terms of the total number of iterations desired in both the previous and new run ('totalsamples'). Therefore, at least one of 'extrait' or 'totalsamples' must not be NULL.

The settings in the new settings object are the same as the previous except:

1) The initial theta value is set to the last sampled value from the previous run

2) The initial rho value is set to the last sampled value from the previous run

3) The data for the initial tree, including the node times, internal sequence and recombination-related latent variables, are set to the last sampled values from the previous run.

Value

Returns an object of class 'pars' with the settings for a sampletrees run that starts where the previous run finished.

Author(s)

Kelly Burkett

References

Burkett KM, McNeney B, Graham J. Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics. 32:1580-2, 2016

Examples

```
#\dontrun{

#system.file("Examples/example_h_pars",package="Rsampletrees")
filename=paste(path.package("Rsampletrees"),"/extdata/example_h_pars",sep="")
runpars=readArgs(filename, check=FALSE)

# Include path in run name so that function can find the necessary files
runname=paste(path.package("Rsampletrees"),"extdata",runpars$RunName, sep="/")
#paste(system.file(package="Rsampletrees"),runpars$RunName, sep="/")
runpars=changeArgs(runpars,RunName=runname)

newpars=restartRun("example-h-2.pars", oldargs=runpars, totalsamples=200000)

#}
```

Index