Package 'Bclim'

February 19, 2015

Type Package

Title Bayesian Palaeoclimate Reconstruction from Pollen

Version 2.3.1

Date 2014-04-15

Author Andrew Parnell, Thinh Doan and James Sweeney

Maintainer Andrew Parnell <Andrew.Parnell@ucd.ie>

Depends MASS, mclust

Imports hdrcde, statmod

Suggests Bchron, doMC, foreach

Description

This package takes pollen/time data from lake cores and produces a Bayesian posterior distribution of palaeoclimate from that location after fitting a non-linear non-Gaussian state-space model.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-04-18 08:39:17

R topics documented:

Bclim-package 2
BclimCompile 3
BclimInterp 5
BclimLayer 6
BclimMCMC 8
BclimMixPar 10
BclimMixSer 12
BclimRun 14
plotBclim 16
plotBclimVol 18

Index **19**

`Bclim-package` *Create palaeoclimate reconstructions from pollen cores*

Description

This package takes pollen core and chronological data to produce palaeoclimate estimates with quantified uncertainties. The package runs in 4 stages, collected together in the function Bclim-Compile.

Details

Package:	Bclim
Type:	Package
Version:	2.3.1
Date:	2013-07-29
License:	GPL >=2

See the function `BclimRun` for a full description of the workings of the package and an example script for getting it to work.

Author(s)

Andrew Parnell, Thinh Doan and James Sweeney

Maintainer: Andrew Parnell <andrew.parnell@ucd.ie>

References

See Arxiv paper at http://arxiv.org/abs/1206.5009, see also references in `BclimLayer`. The webpage http://mathsci.ucd.ie/~parnell_a/Bclim.html contains more detail and background information.

See Also

`BclimRun`, `BclimLayer`, `BclimMixSer`, `BclimMixPar`, `BclimMCMC`, `BclimInterp`, `BclimCompile`, `plotBclim`, `plotBclimVol`

Examples

```
## See help for BclimMCMC and webpage at http://mathsci.ucd.ie/~parnell_a/Bclim.html
```

BclimCompile *Compiles results from other Bclim stages*

Description

This function is designed to compile results from other Bclim stages.

Usage

```
BclimCompile(Layers, Mixtures, MCMC, Interpolations, core.name = "Core")
```

Arguments

Layers	Output from the function BclimLayer
Mixtures	Output from the function BclimMixSer or BclimMixPar
MCMC	Output from the function BclimMCMC
Interpolations	Output from the function BclimInterp
core.name	A character string giving the name of the core

Details

A Bclim run can be separated into 4 stages: 1 - Turning each individual pollen layer into climate marginal data posteriors, 2 - Approximating the marginal data posteriors as mixtures of Gaussians, 3 - Running an MCMC stage to estimate climates and climate volatilities, 4 - Interpolating the climates and volatilities onto a regular grid. This function takes the output from each of these functions and returns a Bclim list object that can be further manipulated.

Value

A list of class Bclim is returned with the following components:

time.grid	The time grid used for interpolation
core.name	The name of the core
clim.interp	The interpolated values of climate for the 3 climate dimensions
vol.interp	The interpolated volatilities for the 3 climate dimensions
MDP	The marginal data posteriors for each layer in the core
ScMean	The raw climate means (used for standardisation purposes)
ScVar	The raw climate variances (used for standardisation purposes)
clim.dims	The names of the different climate dimensions
n	The number of layers in the core
m	The number of climate dimensions (always 3)
n.samp	The number of samples created for the MDPs
Chronsfile	The location of the chronologies file
nchron	The number of chronologies
chron.store	The chronologies used for creating the posterior distributions of climate

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

See Arxiv paper at http://arxiv.org/abs/1206.5009.

See Also

The output here can be used as an input to plotBclim and plotBclimVol. For advanced use of Bclim, see the functions referenced at the top of this file.

Examples

```
## Not run:
# Set the working directory using setwd (not shown)

# Download and load in the response surfaces:
url1 <- 'http://mathsci.ucd.ie/~parnell_a/required.data3D.RData'
download.file(url1,'required_data3D.RData')

# and now the pollen
url2 <- 'http://mathsci.ucd.ie/~parnell_a/SlugganPollen.txt'
download.file(url2,'SlugganPollen.txt')

# and finally the chronologies
url3 <- 'http://mathsci.ucd.ie/~parnell_a/Sluggan_2chrons.txt'
download.file(url3,'Slugganchrons.txt')

# Create variables which state the locations of the pollen and chronologies
pollen.loc <- paste(getwd(),'/SlugganPollen.txt',sep='')
chron.loc <- paste(getwd(),'/Slugganchrons.txt',sep='')

# Load in the response surfaces
load('required.data3D.RData')

## note that all of these functions have further options you can change with
step1 <- BclimLayer(pollen.loc,required.data3D=required.data3D)
step2 <- BclimMixSer(step1)
# See also the parallelised version BclimMixPar if you have doMC and foreach installed
step3 <- BclimMCMC(step2,chron.loc)
# You should probably do some convergence checking after this step
step4 <- BclimInterp(step2,step3)
results <- BclimCompile(step1,step2,step3,step4,core.name="Sluggan Moss")

# Create a plot of MTCO (dim=2)
plotBclim(results,dim=2)

# Create a volatility plot
plotBclimVol(results,dim=2)

## End(Not run)
```

BclimInterp *Interpolation function for producing palaeoclimate histories on a grid*

Description

The function takes output from previous stages of Bclim and produces interpolated values of palaeoclimate (in 3 dimensions) and palaeoclimate volatility.

Usage

```
BclimInterp(Bclim.data, Bclim.res, tgrid = seq(0, 14, by = 0.1))
```

Arguments

Bclim.data	Output from the function BclimMixSer or BclimMixPar
Bclim.res	Output from the function BclimMCMC
tgrid	A grid (not necessarily regular) on which to interpolate climate

Details

A Bclim run can be separated into 4 stages: 1 - Turning each individual pollen layer into climate marginal data posteriors, 2 - Approximating the marginal data posteriors as mixtures of Gaussians, 3 - Running an MCMC stage to estimate climates and climate volatilities, 4 - Interpolating the climates and volatilities onto a regular grid. This function runs stage 4.

Value

Returns a list containing the following elements:

clim.interp	Interpolated values of 3D climate
vol.interp	Interpolated values of 3D climate volatility
time.grid	The time grid used for interpolation

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

See Arxiv paper at http://arxiv.org/abs/1206.5009.

See Also

The main Bclim function is BclimRun. See also the other 3 stages: BclimLayer, BclimMixSer (or BclimMixPar), and BclimMCMC,

Examples

```
## See help for BclimMCMC and webpage at http://mathsci.ucd.ie/~parnell_a/Bclim.html
```

`BclimLayer` *Function to approximate pollen layers as marginal data posteriors*

Description

This function takes a set of pollen data slice by slice and turns it into marginal data posteriors by means of a zero-inflated Negative Binomial model and some response surfaces.

Usage

```
BclimLayer(pollen.loc, required.data3D, nsamples = 1000)
```

Arguments

`pollen.loc`	A character string detailing the location of the 28 taxa pollen file. The 28 taxa should be in the following order: Abies Alnus Betula Carpinus Castanea Cedrus Corylus Ephedra Fagus Juniperus Larix Olea Ostrya Phillyrea Picea Pinus.D Pinus.H Pistacia Quercus.D Quercus.E Salix Tilia Ulmus Artemisia Chenopodiaceae Cyperaceae Ericales Gramineae
`required.data3D`	A list object of response surfaces. A suitable list object can be downloaded as in the example below. You need to load this in before this part of Bclim can be run.
`nsamples`	The number of samples of each marginal data posterior to take. You are unlikely to need to change this from the default value of 1000.

Details

A marginal data posterior (MDP) is of the form pi(ci|yi) where pi is a probability distribution, ci is the 3D climate at depth di and yi is the 28-dimensional pollen vector. This function loops through each layer in the core to produce MDPs which represent the information about climate obtained only from that layer of pollen.

This function could be parallelised to enable speedier computation.

Value

The output is a nsamples x n x m array where nsamples is as above, n is the number of layers and m is the number of climate dimensions (always 3).

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie> and James Sweeney

References

Fore more detail on the algorithm see: Salter-Townshend, M. and J. Haslett (2012). Fast Inversion of a Flexible Regression Model for Multivariate, Zero-Inflated Pollen Counts. Environmetrics. Sweeney, J. (2012). Advances in Bayesian Model Development and Inversion in Multivariate Inverse Inference Problems with application to palaeoclimate reconstruction. Ph. D. thesis, Trinity College Dublin.

See Also

The main Bclim function is BclimRun. See also the other 3 stages: BclimInterp, BclimMixSer (or BclimMixPar), and BclimMCMC,

Examples

```
## Not run:
# Set the working directory using setwd (not shown)

# Download and load in the response surfaces:
url1 <- 'http://mathsci.ucd.ie/~parnell_a/required.data3D.RData'
download.file(url1,'required_data3D.RData')

# and now the pollen
url2 <- 'http://mathsci.ucd.ie/~parnell_a/SlugganPollen.txt'
download.file(url2,'SlugganPollen.txt')

# and finally the chronologies
url3 <- 'http://mathsci.ucd.ie/~parnell_a/Sluggan_2chrons.txt'
download.file(url3,'Slugganchrons.txt')

# Create variables which state the locations of the pollen and chronologies
pollen.loc <- paste(getwd(),'/SlugganPollen.txt',sep='')
chron.loc <- paste(getwd(),'/Slugganchrons.txt',sep='')

# Load in the response surfaces
load('required.data3D.RData')

## note that all of these functions have further options you can change with
step1 <- BclimLayer(pollen.loc,required.data3D=required.data3D)
step2 <- BclimMixSer(step1)
# See also the parallelised version BclimMixPar if you have doMC and foreach installed
step3 <- BclimMCMC(step2,chron.loc)
# You should probably do some convergence checking after this step
step4 <- BclimInterp(step2,step3)
results <- BclimCompile(step1,step2,step3,step4,core.name="Sluggan Moss")

# Create a plot of MTCO (dim=2)
plotBclim(results,dim=2)

# Create a volatility plot
plotBclimVol(results,dim=2)

## End(Not run)
```

BclimMCMC *Bclim Markov chain Monte Carlo run*

Description

A Bclim Markov chain Monte Carlo (MCMC) run to determine the volatilities and climate from a set of marginal data posteriors approximated as mixtures.

Usage

```
BclimMCMC(Bclimdata,chron.loc,nchron=10000,control.mcmc=list(iterations=100000,
burnin=20000,thinby=40,report=100),control.chains=list(v.mh.sd=2,phi1.mh.sd=1,
phi2.mh.sd=10,vstart=statmod::rinvgauss(Bclimdata$n-1,2,1),
Zstart=sample(1:Bclimdata$G,Bclimdata$n,replace=TRUE),phi1start=rep(3,Bclimdata$m),
phi2start=rep(20,Bclimdata$m)),control.priors=list(phi1dlmean=rep(1.275,Bclimdata$m),
phi1dlsd=rep(0.076,Bclimdata$m),phi2dlmean=rep(4.231,Bclimdata$m),
phi2dlsd=rep(0.271,Bclimdata$m)))
```

Arguments

Bclimdata	A set of approximated marginal data posteriors (MDPs) taken from a run of BclimMixPar or BclimMixSer
chron.loc	A character string giving the location of the chronologies file
nchron	The number of chronologies in the chronologies file
control.mcmc	A list containing elements that control the MCMC, including the number of iterations, the size of the burn-in period, the amount to thinby, and how often for the algorithm to report its progress.
control.chains	A list containing elements that control the starting values of the parameters (vstart, Zstart, phi1start and phi2) and the Metropolis-Hastings proposal standard deviation for v, phi1 and phi2.
control.priors	A list containing the prior parameters for the volatilities, given by phi1 and phi2, both of which should be the log-mean and log-sd of the .

Details

This function takes the MDPs, approximated as Gaussian mixtures from BclimMixPar or BclimMixSer, and produces estimated climate and climate volatilities using an MCMC algorithm. The full details are in the Arxiv paper referenced below. The options listed above allow quite a detailed level of control over the behaviour of the algorithm, and convergence should be checked using suitable means (see e.g. the R package boa or coda).

Value

A list object wiht the following elements

v.store	Samples of the posterior estimated volatilities

chron.store	Samples of the used chronologies
c.store	Samples of the posterior estimated climates
z.store	Samples of the posterior mixture indices
phi1	Values used for the IG prior on v for each climate dimension
phi2	Values used for the IG prior on v for each climate dimension
chron.loc	A character string giving the location of the chronology file
nchron	The number of chronologies in the chronology file

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

See Arxiv paper at http://arxiv.org/abs/1206.5009.

See Also

The main Bclim function is BclimRun. See also the other 3 stages: BclimInterp, BclimMixSer (or BclimMixPar), and BclimLayer,

Examples

```
## Not run:
# Set the working directory using setwd (not shown)

# Download and load in the response surfaces:
url1 <- 'http://mathsci.ucd.ie/~parnell_a/media/requireddata3D.RData'
download.file(url1,'required_data3D.RData')

# and now the pollen
url2 <- 'http://mathsci.ucd.ie/~parnell_a/media/SlugganPollen.txt'
download.file(url2,'SlugganPollen.txt')

# and finally the chronologies
url3 <- 'http://mathsci.ucd.ie/~parnell_a/media/Sluggan_2chrons.txt'
download.file(url3,'Slugganchrons.txt')

# Create variables which state the locations of the pollen and chronologies
pollen.loc <- paste(getwd(),'/SlugganPollen.txt',sep='')
chron.loc <- paste(getwd(),'/Slugganchrons.txt',sep='')

# Load in the response surfaces
load('required_data3D.RData')

## note that all of these functions have further options you can change with
step1 <- BclimLayer(pollen.loc,required.data3D=required.data3D)
step2 <- BclimMixSer(step1)
# See also the parallelised version BclimMixPar if you have doMC and foreach installed
step3 <- BclimMCMC(step2,chron.loc)
```

```
# You should probably do some convergence checking after this step
step4 <- BclimInterp(step2,step3)
results <- BclimCompile(step1,step2,step3,step4,core.name="Sluggan Moss")

# Create a plot of MTCO (dim=2)
plotBclim(results,dim=2)

# Create a volatility plot
plotBclimVol(results,dim=2)

## End(Not run)
```

`BclimMixPar` *Parallel version of Bclim mixture analysis*

Description

Function to approximate marginal data posteriors as mixtures of Gaussians

Usage

```
BclimMixPar(MDP, G = 10, num.cores = 4, mixwarnings = FALSE)
```

Arguments

`MDP`	A set of marginal data posteriors, as produced by `BclimLayer`
`G`	The number of Gaussian groups required for each layer to be partitioned into. The default of 10 is usually fine.
`num.cores`	The number of cores to use in the parallel running of this function
`mixwarnings`	Whether to suppress mixture warnings (default) or not.

Details

This function approximates marginal data posteriors (MDPs) as mixtures of Gaussians. The mixture algorithm is taken from the Mclust package which is a required installation for this to run. This is the parallel version, i.e. it uses more than one processor, as opposed to the `BclimMixSer` serial version which will by slower but will run on more machines. For the `BclimMixPar` function to run, you need to have the R packages doMC and foreach installed. Note that this function does not always report output as it runs, depending on the GUI of the system you are on.

Value

Outputs a list containing the following objects:

`MDP`	A nsamples x n x m array (these values are described below)
`n`	The number of layers
`m`	The number of climate dimensions (always 3)

n.samp	The number of samples given in BclimLayer
ScMean	The raw climate means (used for standardisation purposes)
ScVar	The raw climate variances (used for standardisation purposes)
G	The number of mixture groups (as above)
mu.mat	An estimate of the Gaussian mixture mean components
tau.mat	An estimate of the Gaussian mixture precision components
p.mat	An estimate of the Gaussian mixture proportions

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

See Arxiv paper at http://arxiv.org/abs/1206.5009.

See Also

The output here can be used as an input to BclimMCMC. See the main BclimRun function for more details of the other stages you might need to run.

Examples

```
## Not run:
# Set the working directory using setwd (not shown)

# Download and load in the response surfaces:
url1 <- 'http://mathsci.ucd.ie/~parnell_a/required.data3D.RData'
download.file(url1,'required_data3D.RData')

# and now the pollen
url2 <- 'http://mathsci.ucd.ie/~parnell_a/SlugganPollen.txt'
download.file(url2,'SlugganPollen.txt')

# and finally the chronologies
url3 <- 'http://mathsci.ucd.ie/~parnell_a/Sluggan_2chrons.txt'
download.file(url3,'Slugganchrons.txt')

# Create variables which state the locations of the pollen and chronologies
pollen.loc <- paste(getwd(),'/SlugganPollen.txt',sep='')
chron.loc <- paste(getwd(),'/Slugganchrons.txt',sep='')

# Load in the response surfaces
load('required.data3D.RData')

## note that all of these functions have further options you can change with
step1 <- BclimLayer(pollen.loc,required.data3D=required.data3D)
step2 <- BclimMixPar(step1)
# See also the serial version BclimMixSer if you cannot install doMC and foreach
step3 <- BclimMCMC(step2,chron.loc)
```

```
# You should probably do some convergence checking after this step
step4 <- BclimInterp(step2,step3)
results <- BclimCompile(step1,step2,step3,step4,core.name="Sluggan Moss")

# Create a plot of MTCO (dim=2)
plotBclim(results,dim=2)

# Create a volatility plot
plotBclimVol(results,dim=2)

## End(Not run)
```

`BclimMixSer` *Serial version of Bclim mixture analysis*

Description

Function to approximate marginal data posteriors as mixtures of Gaussians

Usage

```
BclimMixSer(MDP, G = 10, mixwarnings = FALSE)
```

Arguments

`MDP`	A set of marginal data posteriors, as produced by `BclimLayer`
`G`	The number of Gaussian groups required for each layer to be partitioned into. The default of 10 is usually fine.
`mixwarnings`	Whether to suppress mixture warnings (default) or not.

Details

This function approximates marginal data posteriors (MDPs) as mixtures of Gaussians. The mixture algorithm is taken from the Mclust package which is a required installation for this to run. This is the serial version, i.e. it only uses one processor, as opposed to the `BclimMixPar` parallel version which will run much faster but requires extra packages to be installed and a multi-core machine.

Value

Outputs a list containing the following objects:

`MDP`	A nsamples x n x m array (these values are described below)
`n`	The number of layers
`m`	The number of climate dimensions (always 3)
`n.samp`	The number of samples given in `BclimLayer`
`ScMean`	The raw climate means (used for standardisation purposes)
`ScVar`	The raw climate variances (used for standardisation purposes)

G	The number of mixture groups (as above)
mu.mat	An estimate of the Gaussian mixture mean components
tau.mat	An estimate of the Gaussian mixture precision components
p.mat	An estimate of the Gaussian mixture proportions

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

See Arxiv paper at http://arxiv.org/abs/1206.5009.

See Also

The output here can be used as an input to BclimMCMC. See the main BclimRun function for more details of the other stages you might need to run.

Examples

```
## Not run:
# Set the working directory using setwd (not shown)

# Download and load in the response surfaces:
url1 <- 'http://mathsci.ucd.ie/~parnell_a/required.data3D.RData'
download.file(url1,'required_data3D.RData')

# and now the pollen
url2 <- 'http://mathsci.ucd.ie/~parnell_a/SlugganPollen.txt'
download.file(url2,'SlugganPollen.txt')

# and finally the chronologies
url3 <- 'http://mathsci.ucd.ie/~parnell_a/Sluggan_2chrons.txt'
download.file(url3,'Slugganchrons.txt')

# Create variables which state the locations of the pollen and chronologies
pollen.loc <- paste(getwd(),'/SlugganPollen.txt',sep='')
chron.loc <- paste(getwd(),'/Slugganchrons.txt',sep='')

# Load in the response surfaces
load('required.data3D.RData')

## note that all of these functions have further options you can change with
step1 <- BclimLayer(pollen.loc,required.data3D=required.data3D)
step2 <- BclimMixSer(step1)
# See also the parallelised version BclimMixPar if you have doMC and foreach installed
step3 <- BclimMCMC(step2,chron.loc)
# You should probably do some convergence checking after this step
step4 <- BclimInterp(step2,step3)
results <- BclimCompile(step1,step2,step3,step4,core.name="Sluggan Moss")
```

```
# Create a plot of MTCO (dim=2)
plotBclim(results,dim=2)

# Create a volatility plot
plotBclimVol(results,dim=2)

## End(Not run)
```

BclimRun *Run all stages of Bclim together*

Description

This function is intended for beggining/light users of Bclim who wish to create palaeoclimate reconstructions from their own pollen data. It collects together the functions BclimLayer, BclimMixSer, BclimMCMC, BclimInterp and BclimCompile.

Usage

```
BclimRun(pollen.loc, chron.loc, core.name = "Core", time.grid = seq(0, 14, length = 100),
  required.data3D = NULL, nchrons = 10000, parallel = FALSE, save.as.you.go = TRUE)
```

Arguments

pollen.loc	A character string detailing the location of the 28 taxa pollen file. The 28 taxa should be in the following order: Abies Alnus Betula Carpinus Castanea Cedrus Corylus Ephedra Fagus Juniperus Larix Olea Ostrya Phillyrea Picea Pinus.D Pinus.H Pistacia Quercus.D Quercus.E Salix Tilia Ulmus Artemisia Chenopodiaceae Cyperaceae Ericales Gramineae
chron.loc	A character string detailing the location of the chronologies file. This file should contain a matrix with the same number of columns as the number of slices in the core (i.e. the number of rows of the pollen data) and at least 2000 rows. Such a file can be created via the Bchron R package from radiocarbon and non-radiocarbon data.
core.name	A character string giving the name of the core. Defaults to 'Core'.
time.grid	A sequence of values of the time variable over which climates and volatilities are to be predicted. Defaults to 0 to 14ka BP with centurial time steps.
required.data3D	A list object of response surfaces. A suitable list object can be downloaded as in the example below. You need to load this in before this part of Bclim can be run.
nchrons	The number of chronologies in the chronologies file. When using output from Bchron this can be safely left at 10,000.
parallel	Whether to run the model using parallel processing technology. To set this as true you must have installed the doMC and foreach R packages.
save.as.you.go	Whether to save output from the differing stages. You should probably keep this set to TRUE in case any of them fail for some reason.

Details

A Bclim run can be separated into 4 stages: 1 - Turning each individual pollen layer into climate marginal data posteriors, 2 - Approximating the marginal data posteriors as mixtures of Gaussians, 3 - Running an MCMC stage to estimate climates and climate volatalities, 4 - Interpolating the climates and volatilities onto a regular grid. This function calls all of these stages in turn and will tidy them up for presentation or use in further analysis, such as for plotting using `plotBclim`. Climate is always 3 dimensional and represented as Growing Degree Days Above 5C (GDD5), The Mean Temperature of the Coldest Month (MTCO) and the ratio of Actual to Potential Evapotranspiration (AET/PET).

Value

A list of class Bclim is returned with the following components:

`time.grid`	The time grid used for interpolation
`core.name`	The name of the core
`clim.interp`	The interpolated values of climate for the 3 climate dimensions
`vol.interp`	The interpolated volatilities for the 3 climate dimensions
`MDP`	The marginal data posteriors for each layer in the core
`ScMean`	The raw climate means (used for standardisation purposes)
`ScVar`	The raw climate variances (used for standardisation purposes)
`clim.dims`	The names of the different climate dimensions
`n`	The number of layers in the core
`m`	The number of climate dimensions (always 3)
`n.samp`	The number of samples created for the MDPs
`Chronsfile`	The location of the chronologies file
`nchron`	The number of chronologies
`chron.store`	The chronologies used for creating the posterior distributions of climate

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

See Arxiv paper at http://arxiv.org/abs/1206.5009.

See Also

The output here can be used as an input to `plotBclim` and `plotBclimVol`. For advanced use of Bclim, see the functions referenced at the top of this file.

Examples

```
## Not run:
# First set the working directory using the setwd command (not shown)

# Download and load in the response surfaces:
url1 <- 'http://mathsci.ucd.ie/~parnell_a/required.data3D.RData'
download.file(url1,'required_data3D.RData')

# and now the pollen
url2 <- 'http://mathsci.ucd.ie/~parnell_a/SlugganPollen.txt'
download.file(url2,'SlugganPollen.txt')

# and finally the chronologies
url3 <- 'http://mathsci.ucd.ie/~parnell_a/Sluggan_2chrons.txt'
download.file(url3,'Slugganchrons.txt')

# Create variables which state the locations of the pollen and chronologies
pollen.loc <- paste(getwd(),'/SlugganPollen.txt',sep='')
chron.loc <- paste(getwd(),'/Slugganchrons.txt',sep='')

# Load in the response surfaces
load('required.data3D.RData')

# Now run Bclim the simple way
RunSluggan <- BclimRun(pollen.loc,chron.loc,core.name="Sluggan Moss",time.grid=seq(0,14,length=100),
  nchrons=10000,required.data3D=required.data3D,parallel=FALSE)

# Create some plots of climate
plotBclim(RunSluggan,dim=1)

# Create some plots of the volatilities
plotBclimVol(RunSluggan,dim=3)

## End(Not run)
```

---

plotBclim *Plots of posterior Bclim climate histories*

Description

Create plots of climate histories from a Bclim run

Usage

```
plotBclim(x, dim = 1, title = NULL, presentleft = TRUE, blob = TRUE, MDPcol = "blue",
  denscol = "red", MDPtransp = 0.1, denstransp = 0.5, leg = TRUE, legloc = "topleft", ...)
```

Arguments

x	The output of a Bclim run from either BclimCompile or BclimRun.
dim	The chosen climate dimension. This could be GDD5 (dim=1), MTCO (dim=2) or AET/PET x 100 (dim=3)
title	The title for the plot
presentleft	ether you wish to have the present on the left hand side (default) or on the right hand side
blob	Whether to include the marginal data posteriors (MDPs) as blobs or not
MDPcol	The colour of the MDPs (only works if blob=TRUE)
denscol	The colour of the 50/75/95% posterior climate ranges
MDPtransp	The transparency (between 0 and 1) of the MDPs
denstransp	The transparency (between 0 and 1) of the posterior climate ranges
leg	Whether to include a legend or not (default TRUE)
legloc	Where to place the legend (e.g. 'topleft', 'bottomright', etc)
...	Other arguments to the plot function

Details

This function creates the default Bclim plots of climate posteriors and MDPs from a Bclim run. If you don't like these then remember that the Bclim object created for x above contains all the information you need to create any other plot you would like.

Value

No output, just a plot

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie> and James Sweeney

See Also

The main Bclim function is BclimRun. See there for all the other information.

Examples

```
## See help for BclimMCMC and webpage at http://mathsci.ucd.ie/~parnell_a/Bclim.html
```

plotBclimVol *Plotting of interpolated climate volatilities*

Description

This function plots volatilities over time for the selected climate dimension after a Bclim run.

Usage

```
plotBclimVol(x, dim = 1, title = NULL, presentleft = TRUE, denscol = "red",
  denstransp = 0.5, leg = TRUE, mean = TRUE, legloc = "topleft", ...)
```

Arguments

x	A Bclim output object either created by BclimCompile or BclimRun.
dim	The chosen climate dimension. This could be GDD5 (dim=1), MTCO (dim=2) or AET/PET x 100 (dim=3)
title	The title for the plot
presentleft	Whether you wish to have the present on the left hand side (default) or on the right hand side
denscol	The colour of the 95% volatility ranges
denstransp	The transparency (between 0 and 1) of the 95% volatility ranges
leg	Whether to include a legend or not (default TRUE)
mean	Whether to include a line indicating the mean volatility
legloc	Where to place the legend (e.g. 'topleft', 'bottomright', etc)
...	Other arguments to the plot function

Value

No output produced, just a plot.

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie> and James Sweeney

See Also

The main Bclim function is BclimRun. See there for all the other information.

Examples

```
## See main function for examples
```

Index

∗Topic **model**
Bclim-package, 2
BclimCompile, 3
BclimInterp, 5
BclimLayer, 6
BclimMCMC, 8
BclimMixPar, 10
BclimMixSer, 12
BclimRun, 14
plotBclim, 16
plotBclimVol, 18
∗Topic **multivariate**
Bclim-package, 2
BclimCompile, 3
BclimInterp, 5
BclimLayer, 6
BclimMCMC, 8
BclimMixPar, 10
BclimMixSer, 12
BclimRun, 14
plotBclim, 16
plotBclimVol, 18
∗Topic **package**
Bclim-package, 2
∗Topic **smooth**
Bclim-package, 2
BclimCompile, 3
BclimInterp, 5
BclimLayer, 6
BclimMCMC, 8
BclimMixPar, 10
BclimMixSer, 12
BclimRun, 14
plotBclim, 16
plotBclimVol, 18

Bclim *(*Bclim-package*)*, 2
Bclim-package, 2
BclimCompile, *2*, 3, *14*, *17, 18*
BclimInterp, *2, 3*, 5, *7*, *9*, *14*
BclimLayer, *2, 3*, *5*, 6, *9–12*, *14*
BclimMCMC, *2, 3*, *5*, *7*, 8, *11*, *13, 14*
BclimMixPar, *2, 3*, *5*, *7–10*, 10, *12*
BclimMixSer, *2, 3*, *5*, *7–10*, 12, *14*
BclimRun, *2*, *5*, *7*, *9*, *11*, *13*, 14, *17, 18*

plotBclim, *2*, *4*, *15*, 16
plotBclimVol, *2*, *4*, *15*, 18