

Package ‘mastif’

October 13, 2022

Type Package

Title Mast Inference and Forecasting

Version 1.0.1

Date 2020-10-19

Author James S. Clark

Maintainer James S. Clark <jimclark@duke.edu>

Description Analyzes production and dispersal of seeds dispersed from trees and recovered in seed traps. Motivated by long-term inventory plots where seed collections are used to infer seed production by each individual plant.

License GPL (>= 2)

Imports Rcpp (>= 0.11.5), RANN, corrplot, xtable, repmis

Depends R (>= 2.10)

LinkingTo Rcpp, RcppArmadillo

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-10-25 16:30:02 UTC

R topics documented:

mastif-package	2
mastClimate	3
mastFillCensus	5
mastif	7
mastMap	11
mastPlot	13
mastPriors	14
mastSim	16

Index	19
--------------	-----------

Description

Seed production is estimated from censuses of trees and seed collections from traps. From locations of known trees and seed traps, infers source strength, coefficients for predictor variables, and parameters for a dispersal kernel. Fecundity is a state-space model allowing for random individual (tree) effects, random year effects and random AR(p) lag effects. Estimates unknown redistribution of seed types to known species identities of trees. Functions begin with 'mast' to avoid conflicts with other packages.

Details

Package: mastif
Type: Package
Version: 1.0.1
Date: 2020-10-19
License: GPL (>= 2)
URL: <http://sites.nicholas.duke.edu/clarklab/code/>

The package mastif estimates fecundity of trees and dispersion of seed observed at seed traps, using information on locations of sources and detectors, and covariates that could explain source strength. Data sets of this type are common and used to understand a range of processes related to seed dispersal, masting, environmental controls on reproduction, sex ratio, and allocation.

Posterior simulation is done by Gibbs sampling. Analysis is done by these functions:

[mastif](#) fits model with Gibbs sampling.

[mastSim](#) simulates data for analysis by mastif.

[mastFillCensus](#) aligns sample years in tree census data with seed trap data.

[mastClimate](#) annotates tree data with covariates for fecundity modeling.

[mastPlot](#) generates plots of the output from mastif.

Author(s)

Author: James S Clark, <jimclark@duke.edu>

References

Clark, JS, C Nunes, and B Tomasek. 2019. Masting as an unreliable resource: spatio-temporal host diversity merged with consumer movement, storage, and diet. *Ecological Monographs*, e01381.

See Also

[mastif](#), [mastSim](#)

A more detailed vignette is can be obtained with:

```
browseVignettes('mastif')
```

mastClimate	<i>Covariates for mast data</i>
-------------	---------------------------------

Description

Annotates treeData for mastif to include covariates.

Usage

```
mastClimate(file, plots, years, months, FUN = 'mean',
            vname = '', lastYear = 2021)
```

Arguments

file	character giving the path and file name where covariate is stored in mastif covariate format (see Details).
plots	character vector from treeData\$plot.
years	numeric vector from treeData\$year.
months	numeric vector of months to include in annual values.
FUN	character specifies how monthly covariate values are combined into annual values.
vname	name to use for a variable in the model that comes from file.
lastYear	last data year to include.

Details

The version of treeData used in mastif can have additional tree years included when there are seed trap years that were not censused or when AR(p) effects extend observations to impute the p years before and after a tree was observed. The function mastFillCensus makes this version of treeData available to the user. The function mastClimate provides a quick way to add plot-year covariates to treeData.

A covariate like minimum monthly temperature is stored in a plot by year_month format, where rownames of file are plot names matching treeData\$plot, and colnames of file could be 2012_1, 2012_2, ... for the 12 months in the year. The numeric vector months holds the months to be included in the annual values, e.g., c(3, 4) for minimum winter temperatures during the period from March through April. To find the minimum for this period, set FUN to 'min'.

More detailed vignettes can be obtained with: browseVignettes('mastif')

Value

A numeric vector equal in length to the number of rows in `treeData` that can be added as a column and included in `formulaFec`.

Author(s)

James S Clark, <jimclark@duke.edu>

References

Clark, J.S., C. Nunes, and B. Tomasek. 2019. Foodwebs based on unreliable foundations: spatio-temporal mastling merged with consumer movement, storage, and diet. *Ecological Monographs*, e01381.

See Also

[mastFillCensus](#) to fill tree census

[mastif](#) for analysis

A more detailed vignette is can be obtained with:

`browseVignettes('mastif')`

website '<http://sites.nicholas.duke.edu/clarklab/code/>'.

Examples

```
d <- "https://github.com/jimclarkatduke/mast/blob/master/liriodendronExample.rData?raw=True"
repmis::source_data(d)

inputs <- list( specNames = specNames, seedNames = seedNames,
               treeData = treeData, seedData = seedData,
               xytree = xytree, xytrap = xytrap)

# interpolate census, add years for AR(p) model
inputs <- mastFillCensus(inputs, p = 3)
treeData <- inputs$treeData #now includes additional years

# include minimum spring temperature of previous year
cfile <- tempfile(fileext = '.csv')
d <- "https://github.com/jimclarkatduke/mast/blob/master/tmin.csv?raw=True"
download.file(d, destfile=cfile)

tyears <- treeData$year - 1
tplots <- treeData$plot

tmp <- mastClimate( file = cfile, plots = tplots,
                  years = tyears, months = 1:4, FUN = 'min')
treeData$tminSprAnomaly <- tmp$x[,3]
inputs$treeData <- treeData

formulaRep <- as.formula( ~ diam )
```

```

formulaFec <- as.formula( ~ diam + tminSprAnomaly )
yearEffect <- list(groups = 'species', p = 3) # AR(3) model, species are lag groups
output <- mastif(inputs = inputs, formulaFec, formulaRep,
                 yearEffect = yearEffect, ng = 2000, burnin = 1000)

```

mastFillCensus	<i>Interpolate census data for seed trap years</i>
----------------	--

Description

Provides interpolated census data to include years when seed data are available. This is used when tree sampling is at a lower frequency than seed-trap collections.

Usage

```

mastFillCensus(inputs, beforeFirst = 15, afterLast = 15, p = 0, verbose = FALSE)

```

Arguments

inputs	list used as input to <code>mastif</code> that must include two character vectors, (<code>specNames</code> , <code>seedNames</code>) and four <code>data.frames</code> (<code>treeData</code> , <code>seedData</code> , <code>xytree</code> , <code>xytrap</code>). See details.
beforeFirst	number of years before a tree is first observed in a census that it should be considered as potentially present.
afterLast	number of years after a tree is last observed in a census that it should be considered as potentially present.
p	if AR(p) model is used (in <code>yearEffect</code>), the number of lag terms.
verbose	if <code>verbose = TRUE</code> information is provided on filling progress.

Details

Masting data sets contain tree census data, held in `treeData`, and seed trap data, held in `seedData`. Most studies monitor seed rain frequently (e.g., annual), while tree censuses occur at intervals of 2 to 5 years. `mastFillCensus` 'fills in' the tree census so that the annual seed data can be used. It is made available to the user so covariates can be added, e.g., with `mastClimate`.

`mastFillCensus` accepts the list of inputs used in `mastif`. The missing years are inserted for each tree with interpolated diameters. `inputs` is returned with objects updated to include the missing census years and modified slightly for analysis by `mastif`.

The function `mastFillCensus` is made accessible to the user, because covariates may be needed for the missing census years. For example, models often include climate variables that change annually. The version of `treeData` returned by `mastFillCensus` can be annotated with additional

columns that can then be included in the model, as specified in `formulaFec`, `formulaRep`, and/or `randomEffect$formulaRan`.

`beforeFirst` and `afterLast` allow the user to control the assumptions about treatment of trees between (and before and after) tree census years. Seed trap data may begin before the first tree census or after the last tree census. Trees may appear in the middle of the study due to ingrowth. They may be lost to mortality. In other words, census data can be left-, right-, and interval-censored.

For the AR(p) model, values are imputed for p years before a tree is first observed and p years after the tree is last observed (`mastif`). These years are inserted by `mastFillCensus`, such that they too can then be annotated with covariate data.

More detailed vignettes can be obtained with:

```
browseVignettes('mastif')
```

Value

`inputs` list includes all inputs to the model, cleaned for `mastif`, with years inserted to `treeData`.

Author(s)

James S Clark, <jimclark@duke.edu>

References

Clark, J.S., C. Nunes, and B. Tomasek. 2019. Foodwebs based on unreliable foundations: spatio-temporal masting merged with consumer movement, storage, and diet. *Ecological Monographs*, e01381.

See Also

`mastSim` simulates data

A more detailed vignette can be obtained with:

```
browseVignettes('mastif')
```

website '<http://sites.nicholas.duke.edu/clarklab/code/>'.

Examples

```
d <- "https://github.com/jimclarkatduke/mast/blob/master/liriodendronExample.rData?raw=True"
repmis::source_data(d)

inputs <- list( specNames = specNames, seedNames = seedNames,
               treeData = treeData, seedData = seedData,
               xytree = xytree, xytrap = xytrap)
inputs <- mastFillCensus(inputs)

formulaFec <- formulaRep <- as.formula(~ diam)

output <- mastif(inputs = inputs, formulaFec, formulaRep, ng = 2000,
```

```
burnin = 1000)
```

 mastif

Gibbs sampler for mast data

Description

Estimates productivity and dispersion of seeds observed at seed traps, using information on locations, and covariates that could explain source strength. Data can be simulated with [mastSim](#).

Usage

```

mastif(inputs, formulaFec=NULL, formulaRep=NULL,
        ng = NULL, burnin = NULL, predList = NULL,
        yearEffect = NULL, randomEffect = NULL,
        modelYears = NULL, plotDims = NULL)

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

## S3 method for class 'mastif'
summary(object, verbose = TRUE, latex = FALSE, ...)
```

Arguments

inputs	list that must include two character vectors, (specNames, seedNames) and four data.frames (treeData, seedData, xytree, xytrap). See details.
formulaFec	R formula for fecundity model, e.g., $\sim \text{diam} + x1$.
formulaRep	R formula for maturation model, e.g., $\sim \text{diam}$.
predList	list holding plots and years for seed prediction.
yearEffect	list holding groups for year effects or AR(p) model.
randomEffect	list holding randGroups and formulaRan for random effects in fecundity.
modelYears	numeric vector can be a subset of years in data set.
ng	numeric number of Gibbs steps.
burnin	numeric number of burnin steps, before predictions are saved.
plotDims	matrix with plot dimensions, used in prediction to determine seed production per plot area. Each row is one plot, with rownames giving the plot name. Columns are 'x1', 'x2', 'y1', 'y2'.
object	currently, also an object of class mastif.
verbose	if TRUE summary sent to screen.
latex	if TRUE tables written to screen in latex format.
x	object of class mastif.
...	further arguments not used here.

Details

`inputs` includes the following:

`specNames` is a character vector containing names of species, `specNames`, that appear in the `treeData$species` column.

`seedNames` is a character vector of seed types that appear as column names in `seedData`.

`treeData` is a `data.frame` holding tree information, including predictors and tree-year identification. Required columns are `plot`, `tree`, `species`, `year`, `diam`, and any other predictors for fecundity or maturation.

`seedData` is a `data.frame` holding seed counts with seed trap and year identification. Required columns are `plot`, `trap`, `year`, and `seedNames`, the latter holding seed counts.

`xytree` is a `data.frame` holding tree locations. Required columns are `plot`, `tree`, `x`, and `y`.

`xytrap` is `data.frame` holding seed trap locations. Required columns are `plot`, `trap`, `x`, and `y`.

`formulaFec` and `formulaRep` specify the models for plant fecundity and maturation. Variables listed in formulas appear as column headings in `treeData`. Note that `formulaFec` and `formulaRep` begin with \sim , not $y \sim$. The response matrix is constructed from seed types in `seedData`.

The `treeData$tree` column has values that are unique for a tree within a plot. These reference the same unique identifiers in `xytree$tree`. In addition to these identifiers, the `data.frame` `xytree` holds columns `x` and `y` for map locations.

The character vector `seedNames` holds the names of columns in `seedData` for seed counts. The elements of `seedNames` are seed types produced by one or more of the species in `specNames`. `seedData` must also include columns for `trap`, `plot`, and `year`, which link with columns in `xytrap`, which additionally includes columns `x` and `y`.

`predList` includes the names of plots and years to be predicted. It can include a numeric value `mapMeters` for the distance between lattice points in the prediction grid. See examples.

`yearEffect` is a list indicating the column names in `treeData` for random groups in year effects or AR(p) models. See examples.

`randomEffect` is a list indicating the column names in `treeData` for random groups in fecundity estimates, the character `randGroups` and the `formulaRan` for random effects. The `formulaRan` must be a subset of predictors from `formulaFec`. See examples.

`modelYears` is a numeric vector of years to include in the analysis.

`ng` is the number of Gibbs steps. `burnin` is the number of initial steps, must be less than `ng`.

Additional arguments to `inputs` can include prior parameters; default values are:

`priorDist = 10` is a prior mean dispersal distance in meters.

`priorVDist = 1` is the prior variance on mean dispersal distance in meters.

`minDist = 2` and `maxDist = 60` are the minimum and maximum values for the mean dispersal kernel in meters.

`minDiam = 2` is the minimum diameter that a tree could be reproductively mature, in cm.

`sigmaMu = .5` and `sigmaWt = nrow(inputs$treeData)` are the prior mean and the prior weight on log fecundity variance.

`maxF = 1e+8`, maximum fecundity, helps stabilize analysis of especially noisy data.

More detailed vignettes can be obtained with:

```
browseVignettes('mastif')
```

Value

Returns an object of `class` "mast", which is a list containing the following components:

inputs	list includes all inputs to the model, including transformations not resulting from posterior simulation.
chains	list of MCMC matrices, each with ng rows, includes: bfec: fecundity coefficients brep: maturation coefficients ugibbs: dispersal parameters sgibbs: residual variance and, after burnin, rmspe (root mean square prediction error) rgibbs: if multiple seed types, then rows are the specNames to seedNames matrix R. bygibbsF, bygibbsR: if yearEffects are included in the model, these are fixed year/lag and random year/lag effects by random group. agibbs: if randomEffects on individuals, rows are covariance matrix.
parameters	list of parameter estimates summarized from chains. acfMat: autocorrelation on fecundity by random group. betaFec: fecundity regression coefficients (log scale). betaRep: maturation regression coefficients (probit scale). dpars: dispersal kernel coefficient, by random group, on meter scale. omegaList: fecundity covariance between trees for the same plot-year. pacfMat: partial autocorrelation on fecundity. pacfSe: standard errors for pacfMat. pacsMat: partial autocorrelation on seed counts. sigma: estimate of residual log fecundity variance. sigmaList: fecundity covariance over years. upars: dispersal kernel coefficient, by random group, on meter ² scale. rMu, rSe: if more than one seedName, posterior mean and standard error on R matrix If yearEffects, then the following will be included in \$parameters: betaYrMu, betaYrSe: posterior mean and standard errors on fixed year effects. betaYrRand, betaYrRandSE: posterior mean and standard errors on random year effects. alphaMu, alphaSe: posterior mean and standard errors on random tree effects. aMu, aSe: posterior mean and standard errors on random effects covariance.
prediction	list of latent variable estimates and prediction: fecPred includes matrEst, fecEstMu, fecEstSe maturation and fecundity estimates and matrPred, fecPred, fecEst maturation and fecundity predictions. seedPred includes seed predictions by species, from estimates of latent fecundity and maturation (<code>_estMean</code> , <code>_estSe</code>) and from the fully generative model (<code>_predMean</code> , <code>_predSe</code>). If predictList is passed to mast, then predictions are returned as seeds per m ² (not per trap) for the regular prediction grid spaced mapMeters apart in the

data.frame seedPredGrid. Tree predictions are returned in treePredGrid are included. If the AR(p) model is used (yearEffect\$p is supplied), then both data.frames include p-yr hind casts and p-yr forecasts.

Author(s)

James S Clark, <jimclark@duke.edu>

References

Clark, J.S., C. Nunes, and B. Tomasek. 2019. Foodwebs based on unreliable foundations: spatio-temporal masting merged with consumer movement, storage, and diet. Ecological Monographs, e01381.

See Also

[mastSim](#) simulates data

A more detailed vignette is can be obtained with:

`browseVignettes('mastif')`

website `'http://sites.nicholas.duke.edu/clarklab/code/'`.

Examples

```
# simulate data (see \link{\code{mastSim}})
seedNames <- specNames <- 'acerRubr'
sim <- list(nyr=10, ntree=20, nplot=5, ntrap=40,
           specNames = specNames, seedNames = seedNames)

inputs <- mastSim(sim)      # simulate data
predList <- list( mapMeters = 3, plots = inputs$plots[1],
                 years = inputs$years )
output <- mastif( inputs = inputs, ng = 3000, burnin = 2000,
                 predList = predList)
mastPlot(output)

# for Liriodendron
d <- "https://github.com/jimclarkatduke/mast/blob/master/liriodendronExample.rData?raw=True"
repmis::source_data(d)

formulaFec <- as.formula( ~ diam ) # fecundity model
formulaRep <- as.formula( ~ diam ) # maturation model

yearEffect <- list(groups = 'species')
randomEffect <- list(randGroups = 'treeID',
                    formulaRan = as.formula( ~ 1 ) )
inputs <- list( specNames = specNames, seedNames = seedNames,
               treeData = treeData, seedData = seedData,
               xytree = xytree, xytrap = xytrap)
output <- mastif(inputs = inputs, formulaFec, formulaRep, ng = 2000,
                 burnin = 1000, yearEffect = yearEffect,
```

```

                                randomEffect = randomEffect)
summary(output)
mastPlot(output)

```

mastMap

Map data and predictions for mastif model

Description

Maps dispersal data (trees and seed traps) with predictions.

Usage

```
mastMap(mapList)
```

Arguments

mapList list that includes data.frames described in the mastif help page: treeData, seedData, xytree, xytrap. Alternatively, mapList can be an object of class mastif simulated by mastSim or fitted by mast. If a class mastif object is used, it must have appended mapPlot (name of plot to map) and mapYears (years to map). These and additional objects in mapList are summarized here:

mapPlot: character for the plot name, which appears in treeData\$plot and seedData\$plot.

mapYears: numeric vector containing years to plot, which appear in treeData\$year and seedData\$year.

treeSymbol: character indicating a column name in treeData holding numeric values that scale the sizes of trees in the map, e.g., diam for diameter.

PREDICT: if TRUE, then predicted seed will be mapped. Must be predicted in mastif.

xlim: vector with upper and lower plot limits.

ylim: vector with upper and lower plot limits.

MAPTRAPS: include seed traps in map.

MAPTREES: include trees in map.

seedMax: numeric value can be used to set maximum seed symbol/prediction contour.

fecMax: numeric value can be used to set maximum fecundity symbol size.

mfrow: numeric vector of length-2 to set arrangement of maps on plot, as in pars().

LEGEND: add legend with species names to map.

plotScale: some level of control on size of plot is possible with values > 1.

treeScale: relative scale for tree symbols, default value = 1.

trapScale: relative scale for trap symbols, default value = 1.

SCALEBAR: add a scale bar with distance to bottom of map.
 scaleValue: numeric value in meters used if SCALEBAR = TRUE.
 mfrow: numeric vector of length-2, giving numbers of rows and columns for plots, see `help(pars)`.
 COLORSCALE: if TRUE, then a color scale is added for predicted seed.

Details

Generates of map of seed traps and trees, with symbols scaled to the sizes relative to seed counts in `sdata$seedNames` and `treeSymbol`. Sizes are adjusted with `scaleTree` and `scaleTrap`.

If `PREDICT = TRUE`, then predictions come in the object fitted in `mastif` with `predictList` used to specify prediction plots and years. See the help page for `mastif`.

More detailed vignettes can be obtained with:

```
browseVignettes('mastif')
```

Value

Only graphical outputs.

Author(s)

James S Clark, <jimclark@duke.edu>

References

Clark, J.S., C. Nunes, and B. Tomasek. 2019. Foodwebs based on unreliable foundations: spatio-temporal masting merged with consumer movement, storage, and diet. *Ecological Monographs*, e01381.

See Also

`mastSim` simulates data

A more detailed vignette is can be obtained with:

```
browseVignettes('mastif')
```

website '<http://sites.nicholas.duke.edu/clarklab/code/>'.

Examples

```
# simulate data (see \link{\code{mastSim}})
seedNames <- specNames <- 'acerRubr'
sim <- list(nyr=10, ntree=30, nplot=5,
           specNames = specNames, seedNames = seedNames)
inputs <- mastSim(sim)

inputs$mapPlot <- 'p1'
inputs$ mapYears = inputs$years[1]
```

```

mastMap( inputs )

# for Pinus
d <- "https://github.com/jimclarkatduke/mast/blob/master/pinusExample.rdata?raw=True"
repmis::source_data(d)

specNames <- c("pinuEchi","pinuRigi","pinuStro","pinuTaed","pinuVirg")
seedNames <- c(specNames, "pinuUNKN")

mapList <- list( treeData = treeData, seedData = seedData,
                 specNames = specNames, seedNames = seedNames,
                 xytree = xytree, xytrap = xytrap, mapPlot = 'DUKE_BW',
                 mapYears = c(2004:2007), treeScale = .5, trapScale=1.2,
                 plotScale = 1.2, LEGEND=TRUE)

mastMap(mapList)

```

mastPlot

Plot mast model

Description

Plots data fitted with `mastif` in package `mastif`.

Usage

```
mastPlot(output, plotPars = NULL)
```

Arguments

<code>output</code>	list of class <code>mast</code> .
<code>plotPars</code>	list of inputs for plotting can include: <code>outFolder</code> : if <code>SAVEPLOTS = TRUE</code> path to saved files. <code>SAVEPLOTS = FALSE</code> : if <code>TRUE</code> writes plots to <code>.pdf</code> files. <code>trueValues</code> : if passed from <code>mastSim</code> output then <code>trueValues</code> will be included in plots. <code>RMD</code> : if set to <code>"pdf"</code> or <code>"html"</code> writes output to R markdown file in working directory.

Details

If `SAVEPLOTS = TRUE` plots are saved to files in `outFolder`. If `RMD = "pdf"`, output is written to a R markdown file that can be edited and knitted. Maps are not included in this option. Otherwise, plots are rendered to the screen.

More detailed vignettes can be obtained with:

```
browseVignettes('mastif')
```

Value

Currently, there are graphical outputs.

Author(s)

James S Clark, <jimclark@duke.edu>

References

Clark, J.S., C. Nunes, and B. Tomasek. 2019. Foodwebs based on unreliable foundations: spatio-temporal masting merged with consumer movement, storage, and diet. *Ecological Monographs*, e01381.

See Also

[mastSim](#) simulates data

A more detailed vignette is can be obtained with:

```
browseVignettes('mast')
```

website '<http://sites.nicholas.duke.edu/clarklab/code/>'.

Examples

```
# simulate data
seedNames <- specNames <- 'acerRubr'
sim <- list(nyr=10, ntree=30, nplot=5, specNames = specNames, seedNames = seedNames)
inputs <- mastSim(sim)
output <- mastif( inputs = inputs, ng = 4000, burnin = 2000 )

plotPars <- list(trueValues = inputs>trueValues)
mastPlot(output, plotPars)
```

mastPriors

Obtain prior parameter values for mastif from file

Description

Prior parameter values may be saved in a file by species or by genus. `mastPriors` looks for a species-level prior first. If not found, it can substitute a genus-level prior.

Usage

```
mastPriors(file, specNames, code, genus = 'NULL')
```

Arguments

file	character giving the path and file name where parameters are stored (see Details).
specNames	character vector of names in treeData\$species.
code	character name of column holding specNames in parameter file.
genus	character name of genus held in "genus" column of parameter file.

Details

The file includes rows with genera, given in column "genus", or "species". Species rows also have an entry for genus, with the species code given in the column named code. Additional columns are names of prior parameters, including:

priorDist: mean parameter for dispersal kernel (m), related to kernel parameter u as $d \leftarrow \pi \cdot \sqrt{u} / 2$. The estimated values for these parameters are found in `output$parameters$upars` and `output$parameters$dparams`, where `output` is an object fitted by `mastif`.

minDist: the lower bound for the mean parameter d of the dispersal kernel (m).

maxDist: the upper bound for the mean parameter d of the dispersal kernel (m).

priorVDist: variance on the mean parameter for dispersal kernel (m^2). For large values, the prior distribution of d (and by variable change, u) becomes `dunif(d, minDist, maxDist)`.

minDiam: below this diameter trees of unknown status are assumed immature (cm).

maxDiam: above this diameter trees of unknown status are assumed mature (cm).

maxFec: maximum seeds per tree per year

More detailed vignettes can be obtained with: `browseVignettes('mastif')`

Value

A data.frame with a row for each specNames and columns for prior parameter values. Where file contains species-level parameter values, they will be used. If a separate row in file holds genus-level parameters, with the entry for code == 'NA', then genus-level parameters will be substituted. In other words, these genus rows are default values.

Author(s)

James S Clark, <jimclark@duke.edu>

References

Clark, J.S., C. Nunes, and B. Tomasek. 2019. Foodwebs based on unreliable foundations: spatio-temporal masting merged with consumer movement, storage, and diet. *Ecological Monographs*, e01381.

See Also

[mastFillCensus](#) to fill tree census

[mastif](#) for analysis

A more detailed vignette is can be obtained with:

`browseVignettes('mastif')`

website `'http://sites.nicholas.duke.edu/clarklab/code/'`.

Examples

```
# data
d <- "https://github.com/jimclarkatduke/mast/blob/master/pinusExample.rdata?raw=True"
repmis::source_data(d)

# prior parameter values
pfile <- tempfile(fileext = '.txt')
d <- "https://github.com/jimclarkatduke/mast/blob/master/priorParameters.txt?raw=True"
download.file(d, destfile = pfile)

specNames <- c("pinuEchi", "pinuRigi", "pinuStro", "pinuTaed", "pinuVirg")
seedNames <- c(specNames, "pinuUNKN")

priorTable <- mastPriors(file = pfile, specNames,
                        code = 'code4', genus = 'pinus')

inputs <- list( specNames = specNames, seedNames = seedNames,
                treeData = treeData, seedData = seedData,
                xytree = xytree, xytrap = xytrap,
                priorTable = priorTable, seedTraits = seedTraits)

formulaRep <- as.formula( ~ diam )
formulaFec <- as.formula( ~ diam )

output <- mastif(inputs = inputs, formulaFec, formulaRep,
                ng = 2000, burnin = 1000)
```

mastSim

Data simulation for mast model

Description

Simulates data for analysis by [mastif](#) in package `mastif`.

Usage

```
mastSim(sim)
```

Arguments

sim list of inputs for simulation, see Details.

Details

The list `sim` contains the following:

`specNames`: character vector of species names.

`seedNames`: character vector of seed names.

`nyr = 5`: average number of years for a plot

`ntree = 10`: average number of trees in `specNames` on a plot

`plotWide = 100`: diameter of plot

`nplot = 3`: number of plots

`ntrap = 20`: average number of seed traps on a plot

`meanDist = 25`: mean dispersal (meters)

Value

Returns an object of `class "mastif"`, a list containing the following components:

<code>distall</code>	matrix of distances between all seed traps (rows) and trees (columns). Block diagonal structure by plot, padded with the value 10000 for tree/trap combinations on different plots.
<code>distall</code>	seed trap by tree matrix of distances.
<code>formulaFec</code>	formula for the fecundity model.
<code>formulaRep</code>	formula for the maturation model.
<code>plots</code>	character vector of plot names.
<code>R</code>	species to seed type matrix.
<code>seedData</code>	data.frame of seed data needed for <code>mastif</code> .
<code>seedNames</code>	character vector of seed types.
<code>sim</code>	inputs to <code>mastSim</code> .
<code>specNames</code>	character vector of species.
<code>treeData</code>	data.frame of tree data needed for <code>mastif</code> .
<code>trueValues</code>	list holding true parameter values includes: <code>fec</code> : vector of log fecundity states. <code>repr</code> : vector of maturation states. <code>betaFec</code> : fecundity parameter values. <code>betaRep</code> : maturation parameter values. <code>upar</code> : dispersal parameter, m^2 .
<code>xytrap</code>	data.frame of seed locations needed for <code>mastif</code> .
<code>xytree</code>	data.frame of tree locations needed for <code>mastif</code> .
<code>years</code>	numeric vector years.

Author(s)

James S Clark, <jimclark@duke.edu>

References

Clark, J.S., C. Nunes, and B. Tomasek. 2019. Foodwebs based on unreliable foundations: spatio-temporal masting merged with consumer movement, storage, and diet. *Ecological Monographs*, in press.

See Also

`mastSim` simulates data

A more detailed vignette is can be obtained with:

`browseVignettes('mastif')`

website '<http://sites.nicholas.duke.edu/clarklab/code/>'.

Examples

```
# simulate data
seedNames <- specNames <- 'acerRubr'
sim <- list(nyr = 10, ntree = 30, nplot = 5, ntrap = 40,
           specNames = specNames, seedNames = seedNames)
inputs <- mastSim(sim)
output <- mastif( inputs = inputs, ng = 500, burnin = 200 )

# increase iterations, then plot:

output <- mastif( inputs = output, ng = 2000, burnin = 1000 )
plotPars <- list(trueValues = inputs>trueValues)
mastPlot(output, plotPars)
```

Index

- * **Bayesian analysis**
 - mastif-package, 2
 - * **Gibbs sampling**
 - mastif-package, 2
 - * **Hamiltonian Markov chain**
 - mastif-package, 2
 - * **autoregressive model**
 - mastif-package, 2
 - * **ecology**
 - mastif-package, 2
 - * **hierarchical Bayes**
 - mastif-package, 2
 - * **masting**
 - mastif-package, 2
 - * **package**
 - mastif-package, 2
 - * **prediction**
 - mastif-package, 2
 - * **seed dispersal**
 - mastif-package, 2
 - * **seed production**
 - mastif-package, 2
 - * **source detection**
 - mastif-package, 2
 - * **state-space model**
 - mastif-package, 2
 - * **tree fecundity**
 - mastif-package, 2
- class, 9, 17
- mastClimate, 2, 3, 5
mastFillCensus, 2, 4, 5, 16
mastif, 2–6, 7, 13, 16
mastif-package, 2
mastMap, 11
mastPlot, 2, 13
mastPriors, 14
mastSim, 2, 3, 6, 7, 10, 12, 14, 16, 18
- print.mastif(mastif), 7
summary.mastif(mastif), 7