Package ‘hierarchicalDS’

October 13, 2022

Maintainer Paul B Conn <paul.conn@noaa.gov>

License Unlimited

Title Functions to Perform Hierarchical Analysis of Distance Sampling Data

Type Package

LazyLoad yes

Author P.B. Conn \email{paul.conn@@noaa.gov}

Description Functions for performing hierarchical analysis of distance sampling data, with ability to use an areal spatial ICAR model on top of user supplied covariates to get at variation in abundance intensity. The detection model can be specified as a function of observer and individual covariates, where a parametric model is supposed for the population level distribution of covariate values. The model uses data augmentation and a reversible jump MCMC algorithm to sample animals that were never observed. Also included is the ability to include point independence (increasing correlation multiple observer's observations as a function of distance, with independence assumed for distance=0 or first distance bin), as well as the ability to model species misclassification rates using a multinomial logit formulation on data from double observers. There is also the the ability to include zero inflation, but this is only recommended for cases where sample sizes and spatial coverage of the survey are high.

Version 3.0

Imports truncnorm, mvtnorm, Matrix, coda, xtable, mc2d, ggplot2, rgeos, MCMCpack

Date 2019-07-01

RoxygenNote 6.1.1

NeedsCompilation no

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2019-07-02 18:10:03 UTC

R topics documented:

`calc_linex_a` *estimate optimal 'a' parameter for linex loss function*

Description

estimate optimal 'a' parameter for linex loss function

Usage

```
calc_linex_a(Pred.G, Obs.G, min.a = 1e-05, max.a = 1)
```

Arguments

`Pred.G`	Predicted group abundance
`Obs.G`	Observed group abundance
`min.a`	Minimum value for linex 'a' parameter
`max.a`	Maximum value for linex 'a' parameter

Value

The optimal tuning parameter for linex loss function as determined by minimum sum of squares

Author(s)

Paul B. Conn

`convert.HDS.to.mcmc` *function to convert HierarchicalDS MCMC list vector (used in estimation) into an mcmc object (cf. coda package)*

Description

function to convert HierarchicalDS MCMC list vector (used in estimation) into an mcmc object (cf. coda package)

Usage

```
convert.HDS.to.mcmc(MCMC, N.hab.pois.par, N.hab.bern.par, n.ind.cov,
  n.cov.cols, Hab.pois.names, Hab.bern.names, Det.names, Cov.names,
  MisID.names, N.par.misID = NULL, misID.mat = NULL,
  fix.tau.nu = FALSE, misID = TRUE, spat.ind = TRUE,
  point.ind = TRUE)
```

Arguments

`MCMC`	list vector providing MCMC samples for each parameter type
`N.hab.pois.par`	see help for mcmc_ds.R
`N.hab.bern.par`	see help for mcmc_ds.R
`n.ind.cov`	see help for mcmc_ds.R
`n.cov.cols`	number of columns in Cov.prior.parms matrix
`Hab.pois.names`	see help for mcmc_ds.R
`Hab.bern.names`	see help for mcmc_ds.R
`Det.names`	see help for mcmc_ds.R
`Cov.names`	see help for mcmc_ds.R
`MisID.names`	see help for mcmc_ds.R

`N.par.misID`	see help for mcmc_ds.R
`misID.mat`	see help for mcmc_ds.R
`fix.tau.nu`	see help for mcmc_ds.R
`misID`	see help for mcmc_ds.R
`spat.ind`	see help for mcmc_ds.R
`point.ind`	see help for mcmc_ds.R

Author(s)

Paul B. Conn

`generate_inits` *generate initial values for MCMC chain if not already specified by user*

Description

generate initial values for MCMC chain if not already specified by user

Usage

```
generate_inits(DM.hab, DM.det, G.transect, Area.trans, Area.hab, Mapping,
  point.ind, spat.ind, grp.mean)
```

Arguments

`DM.hab`	design matrix for habitat model
`DM.det`	design matrix for detection model
`G.transect`	a vector of the number of groups of animals in area covered by each transect
`Area.trans`	a vector giving the proportion of a strata covered by each transect
`Area.hab`	a vector of the relative areas of each strata
`Mapping`	a vector mapping each transect to it's associated strata
`point.ind`	is point independence assumed (TRUE/FALSE)
`spat.ind`	is spatial independence assumed? (TRUE/FALSE)
`grp.mean`	pois1 parameter for group size

Value

a list of initial parameter values

Author(s)

Paul B. Conn

`generate_inits_misID` *generate initial values for misID model if not already specified by user*

Description

generate initial values for misID model if not already specified by user

Usage

```
generate_inits_misID(DM.hab.pois, DM.hab.bern, DM.det, N.hab.pois.par,
  N.hab.bern.par, G.transect, Area.trans, Area.hab, Mapping, point.ind,
  spat.ind, grp.mean, misID, misID.mat, N.par.misID)
```

Arguments

`DM.hab.pois`	a list of design matrices for the Poisson habitat model (elements are named sp1,sp2, etc.)
`DM.hab.bern`	If a hurdle model, a list of design matrices for the Bernoulli habitat model (elements are named sp1,sp2, etc.) (NULL if not hurdle)
`DM.det`	design matrix for detection model
`N.hab.pois.par`	vector giving number of parameters in the Poisson habitat model for each species
`N.hab.bern.par`	vector giving number of parameters in the Bernoulli habitat model for each species (NULL if not hurdle)
`G.transect`	a matrix of the number of groups of animals in area covered by each transect; each row gives a separate species
`Area.trans`	a vector giving the proportion of a strata covered by each transect
`Area.hab`	a vector of the relative areas of each strata
`Mapping`	a vector mapping each transect to it's associated strata
`point.ind`	is point independence assumed (TRUE/FALSE)
`spat.ind`	is spatial independence assumed? (TRUE/FALSE)
`grp.mean`	a vector giving the pois1 parameter for group size (one entry for each species)
`misID`	if TRUE, indicates that misidentification is incorporated into modeling
`misID.mat`	a matrix specifying which elements of the misID matrix are linked to model equations
`N.par.misID`	a vector giving the number of parameters for each misID model (in multinomial logit space)

Value

a list of initial parameter values

Author(s)

Paul B. Conn

`get_confusion_array` *Fill confusion array - one confusion matrix for each individual (DEPRECATED)*

Description

Fill confusion array - one confusion matrix for each individual (DEPRECATED)

Usage

```
get_confusion_array(Confuse, Cov = NULL, Beta, n.indiv, misID.mat,
  misID.formulas, symm = TRUE)
```

Arguments

`Confuse`	An 3-dimensional array, with dimensions (# of individuals, # of rows in misID.mat, # of cols of misID.mat)
`Cov`	Data frame including all covariates for the misclassification model (individuals are on rows)
`Beta`	A list where each entry is a vector giving the parameters of the misID model
`n.indiv`	Integer giving the number of individuals
`misID.mat`	With true state on rows and assigned state on column, each positive entry provides an index to misID.models (i.e. what model to assume on multinomial logit space); a 0 indicates an impossible assigment; a negative number designates which column is to be obtained via subtraction
`misID.formulas`	A formula vector providing linear model-type formulas for each positive value of misID.mat. If the same model is used in multiple columns it is assumed that all fixed effects (except the intercept) are shared
`symm`	if TRUE, symmetric classification probabilities are applied (e.g. pi^12=pi^21)

Value

A filled version of Confuse

Author(s)

Paul B. Conn

`get_confusion_mat` *Fill a list with confusion matrices for each record*

Description

Fill a list with confusion matrices for each record

Usage

```
get_confusion_mat(Cur.dat, Beta, misID.mat, misID.models,
  misID.symm = TRUE, stacked.names, factor.ind, Levels)
```

Arguments

`Cur.dat`	Matrix giving data (records and covariates) - multiple rows can be given (e.g. reflecting different observers)
`Beta`	A list where each entry is a vector giving the parameters of the misID model
`misID.mat`	With true state on rows and assigned state on column, each positive entry provides an index to misID.models (i.e. what model to assume on multinomial logit space); a 0 indicates an impossible assigment; a negative number designates which column is to be obtained via subtraction
`misID.models`	A formula vector providing linear model-type formulas for each positive value of misID.mat. If the same model is used in multiple columns it is assumed that all fixed effects (except the intercept) are shared
`misID.symm`	if TRUE, symmetric classification probabilities are applied (e.g. pi^12=pi^21)
`stacked.names`	A character vector giving column names for the data
`factor.ind`	An integer vector holding whehter each column of data is to be treated as numeric or factor
`Levels`	A list, each entry of which corresponds to a column name for factor variables and gives the possible levels of those factors

Value

A list of confusion matrices, one for each row in Cur.dat

Author(s)

Paul B. Conn

`get_mod_matrix` *function to produce a design matrix given a dataset and user-specified formula object*

Description

function to produce a design matrix given a dataset and user-specified formula object

Usage

```
get_mod_matrix(Cur.dat, stacked.names, factor.ind, Det.formula, Levels)
```

Arguments

`Cur.dat`	current dataset
`stacked.names`	column names for current dataset
`factor.ind`	a list of indicator variables (1 = factor/categorical variable, 0 = continuous variable)
`Det.formula`	a formula object
`Levels`	A list object giving the number of levels for factor variables

Value

a design matrix

Author(s)

Paul B. Conn

`hierarchical_DS` *Primary function for hierarchical, areal analysis of distance sampling data (without movement). This function pre-processes data and calls other functions to perform the analysis, and is the only function the user needs to call themselves.*

Description

Primary function for hierarchical, areal analysis of distance sampling data (without movement). This function pre-processes data and calls other functions to perform the analysis, and is the only function the user needs to call themselves.

Usage

```
hierarchical_DS(Dat, Adj, Area.hab = 1, Mapping, Area.trans, Observers,
  Bin.length, Hab.cov, Obs.cov, Hab.pois.formula,
  Hab.bern.formula = NULL, Det.formula, detect = TRUE, Cov.prior.pdf,
  Cov.prior.parms, Cov.prior.fixed, Cov.prior.n, n.obs.cov = 0,
  pol.eff = c(1:2), ZIP = FALSE, point.ind = TRUE,
  spat.ind = FALSE, last.ind = FALSE, cor.const = FALSE,
  fix.tau.nu = FALSE, srr = TRUE, srr.tol = 0.5, misID = FALSE,
  misID.models = NULL, misID.mat = NULL, misID.symm = TRUE,
  Inits = NULL, grps = FALSE, M, Control, adapt = TRUE, Prior.pars,
  post.loss = TRUE)
```

Arguments

`Dat`	A data frame with the following columns: (1)transect ID; (2)match number currently, a maximum of 2 observers on each transect; (3)(Observer ID); (4)(Observation (0/1)); (5) Observed species (integer - the max integer being 'unknown' if applicable) [NOTE: modeled as factor, but need to be input as integers to account for unknown species observations] (6-x)(Observer covariates); (things like survey conditions or observer skill; things that don't change during a transect. Note these also need to be provided in Obs.cov) (x+1)(Distance; if all integers, assumed to be discrete bins; if continuous, assumed standardized to (0,1) interval); (x+2-??)(Group size and other individual covariates thought to influence detection; if group size is one of them, it's assumed to be column x+2); Note that column names can be used to tag covariates, and that object types (e.g. numeric, factor) will be preserved in analysis
`Adj`	Adjacency matrix for habitat cells (diagonal matrix implies spatial independence)
`Area.hab`	A vector giving the area of each geographical strata (default is equal area)
`Mapping`	A vector giving the habitat cell id number for each transect
`Area.trans`	A vector giving the effective area covered by each transect as fraction of total area in the strata it is located
`Observers`	A (2 x number of transects) matrix giving the observers IDs that were present for each transect (the 2nd row is to contain NAs if only 1 observer was present)
`Bin.length`	If distances are binned, this vector gives the relative length of each distance bin (vector must sum to one)
`Hab.cov`	A data.frame object giving covariates thought to influence abundance intensity at strata level; column names index individual covariates
`Obs.cov`	A (max # of observers X # of transects X # of observer covariates) size array giving observer covariate values for each transect flown
`Hab.pois.formula`	A list of formulas (one for each species) giving the specific model for Poisson abundance intensity at the strata level (e.g., ~Vegetation+Latitude) for each species

`Hab.bern.formula`	If ZIP=TRUE, a list of formulas (one for each species) giving the specific model for the zero component for abundance intensity at the strata level (e.g., ~Vegetation+Latitude) for each species
`Det.formula`	A formula giving the model for detection probability (e.g. ~Distance+Group+Visibility+Observer). Note that there are several "reserved" variable names. "Distance", "Observer", "Species", and "Group" are reserved variable names.
`detect`	If TRUE (the default), detectability is estimated; if FALSE, assumes detection probability is 1.0 (i.e. a strip transect with perfect detection).
`Cov.prior.pdf`	If individual covariates are provided, this character matrix gives the form of the prior pdfs for each covariate current possibilities are "poisson", "pois1","poisson_ln","pois1_ln", and "bernoulli". "pois1" is 1+x where x~poisson; "poisson_ln" and "pois1_ln" are lognormal poisson models that incorporate overdispersion. Note the dimension of this matrix are (# species X # of covariates)
`Cov.prior.parms`	A (s X k X n) array where s is the number of species, n is the number of individual covariates (other than distance), and k is the maximum number of parameters considered for a single covariate (NAs can be used to fill this matrix out for covariate priors that have <k parameters). If Cov.prior.fixed=1 for a given entry, the prior parameters supplied in each column apply to the prior pdf itself, and are treated as fixed. If Cov.prior.fixed=0, the model will attempt to estimate the posterior distribution of model parameters, given hyperpriors. In this case, it is actually the hyperpriors that are being specified. For "poisson", and "pois1", it is assumed that lambda~gamma(alpha,beta), so alpha and beta must be supplied. For "poisson_ln", and "pois1_ln", the model is lambda_i=exp(theta + epsilon_i), where epsilon_i has a N(0,sigma^2 distribution) so it is priors for theta and sigma that are specified (in that order). Theta is assumed to have a normal(mu,s^2) distribution, and sigma is assumed to have a uniform(0,a) distribution; thus, priors are specified for these models as (mu,s, and a). For Bernoulli covariates, the prior is assumed to be Beta(a,b), so two values must be supplied (a and b).
`Cov.prior.fixed`	An indicator matrix specifying which (if any) individual covariate distributions should be fixed during estimation
`Cov.prior.n`	An (# species X # indiv. covariates) matrix giving the number of parameters in each covariate pdf
`n.obs.cov`	Number of observer covariates (e.g., seat position, visibility, etc.)
`pol.eff`	For continuous distance, which polynomial degrees to model (default is c(1:2); an intercept is always estimated when "Distance" is listed in "Det.formula")
`ZIP`	If TRUE, estimate ZIP model for abundance that includes a Bernoulli model for zeros and a Poisson + 1 model for positive values (default is FALSE)
`point.ind`	Estimate a correlation parameter for detection probability that's an increasing function of distance?
`spat.ind`	If TRUE, assumes spatial independence (no spatial random effects on abundance intensity) default is FALSE

`last.ind`	If point independence is modeled (point.ind=TRUE), last.ind=TRUE will set observer covariance to zero for the greatest distance and maximal correlation in first bin (default is FALSE)
`cor.const`	If TRUE, forces estimates of correlation associated with point independence to be positive if last.ind==FALSE or negative if last.ind==TRUE (default is FALSE)
`fix.tau.nu`	If TRUE, fixes tau.nu during estimation (the value to fix it to can be provided in "Inits")
`srr`	If TRUE, uses spatially retricted regression, where smoothing occurs on residuals and all spatial effects are orthogonal to the linear predictors (by default, analysis is limited to the highest 50 eigenvalues of the decomposition of the residual projection matrix to reduce computing time)
`srr.tol`	Threshold eigenvalue level for SRR; only eigenvectors with higher eigenvalues than srr.tol are included in SRR formulation (default is 0.5)
`misID`	If TRUE, updates species for observed animals and estimates misID parameters
`misID.models`	A formula vector providing linear model-type formulas for each positive value of misID.mat.
`misID.mat`	With true state on rows and assigned state on column, each positive entry provides an index to misID.models (i.e. what model to assume on multinomial logit space); a 0 indicates an impossible assigment; a negative number designates which column is to be obtained via subtraction
`misID.symm`	If TRUE, the constraint pi^ilj=pi^jli is implemented; in this case, entries for pi^jli are all assumed to be given a ’-1’ in misID.mat
`Inits`	An (optional) list object providing initial values for model parameters, with the following objects: "hab.pois": Initial values for habitat linear predictor parameters for poisson model; "hab.bern": If ZIP=TRUE, initial values for habitat linear predictor parameters for bernoulli zero model; "det": Initial values for detection model (includes distance, observer, env. variables, and individual covariates); "cor.par": If point.ind==TRUE, this is an initial value for the correlation parameter (which must be in (0,1)); "Nu": Gives log(lambda) for each spatial strata; "Eta.pois": If spat.ind==FALSE, spatial random effects for Poisson abundance model; one for each cell and for each species "Eta.bern": If spat.ind==FALSE & ZIP=TRUE, spatial random effects for Bernoulli abundance model; one for each cell and for each species "tau.eta.pois": If spat.ind==FALSE, precision for spatial ICAR model(s) for the Poisson component "tau.eta.bern": If spat.ind==FALSE & ZIP=TRUE, precision for spatial ICAR model(s) for the Bernoulli component "tau.nu": Precision for Nu (overdispersion relative to the Poisson distribution) One need not specify an initial value for all parameter types (if less are specified, the others are generated randomly)
`grps`	If FALSE, detections are assumed to all be of individual animals
`M`	Matrix with species-specific rows giving maximum possible value for number of groups present in each transect (in practice just set high enough that values at M and above are never sampled during MCMC) and can be fine tuned as needed
`Control`	A list object including the following objects: "iter": number of MCMC iterations; "burnin": number of MCMC burnin iterations; "thin": if specified,

	how many iterations to skip between recorded posterior samples; "adapt": if adapt==TRUE, this gives the number of additional MCMC iterations should be performed to adapt MCMC proposals to optimal ranges prior to final MCMC run; "MH.cor": Metropolis-hastings tuning parameter for updating the correlation parameter (if point.ind==TRUE); "MH.nu": MH tuning parameters for Nu parameters (dimension = # species X # of unique strata sampled) "RJ.N": A matrix giving the maximum number of additions and deletions proposed in an iteration of the RJMCMC algorithm for each species (row) and each transect (column) "iter.fix.N": Number of iterations to skip RJMCMC step at beginning of estimation (useful for cases when estimation is unstable)
`adapt`	If adapt==TRUE, run an additional Control$adapt number of MCMC iterations to optimize MCMC proposal distributions prior to primary MCMC
`Prior.pars`	A list object giving parameters of prior distribution. Includes the following objects "a.eta": alpha parameter for prior precision of spatial process (assumed Gamma(a.eta,b.eta)) "b.eta": beta parameter for prior precision of spatial process (assumed Gamma(a.eta,b.eta)) "a.nu": alpha parameter for prior precision of overdispersion process (assumed Gamma(a.nu,b.nu)) "b.nu": beta parameter for prior precision of overdispersion process (assumed Gamma(a.nu,b.nu)) "beta.tau": prior precision for regression coefficients (assumed Normal(0,(beta.tau*X'X)^(-1))
`post.loss`	If TRUE, calculates observed values and posterior predictions for detection data to use with posterior predictive loss functions

Value

returns a list with the following objecs: MCMC: A list object containing posterior samples; Accept: A list object indicating the number of proposals that were accepted for parameters updated via Metropolis-Hastings; Control: A list object giving MCMC tuning parameters (which are updated if the 'adapt' alorithm is used)

Author(s)

Paul B. Conn <`paul.conn@noaa.gov`>

Examples

```
print("example analysis included in the script example_analysis.R")
```

`linear_adj` *Produce an adjacency matrix for a vector*

Description

Produce an adjacency matrix for a vector

Usage

```
linear_adj(x)
```

Arguments

`x`	length of vector

Value

adjacency matrix

Author(s)

Paul Conn

`log_lambda_gradient` *compute the first derivative of log_lambda likelihood component for Langevin-Hastings*

Description

compute the first derivative of log_lambda likelihood component for Langevin-Hastings

Usage

```
log_lambda_gradient(Mu, Nu, Sampled, Area, N, var.nu)
```

Arguments

`Mu`	expected value for all cells
`Nu`	current observed valus (all cells)
`Sampled`	Vector giving the cell identities for all sampled cells
`Area`	Proportional area of each sampled cell that is covered by one or more transects
`N`	number of groups in each transect
`var.nu`	variance of the overdispersion process

Value

a gradient value

Author(s)

Paul B. Conn

`log_lambda_log_likelihood` *compute the likelihood for nu parameters*

Description

compute the likelihood for nu parameters

Usage

```
log_lambda_log_likelihood(Log.lambda, DM, Beta, Eta = 0, SD, N, Sampled,
  Area)
```

Arguments

`Log.lambda`	Log of poisson intensities for total areas sampled in each sampled strata
`DM`	the design matrix
`Beta`	linear predictor parameters for the log of abundance intensity
`Eta`	a vector of spatial random effects
`SD`	standard deviation of the overdispersion process
`N`	a vector giving the current iteration's number of groups in the area
`Sampled`	Index for which cells were actually sampled
`Area`	Total area sampled in each sampled cell

Value

the log likelihood associated with the data and the current set of parameters

Author(s)

Paul B. Conn

`mcmc_ds` *Function for MCMC analysis*

Description

Function for MCMC analysis

Usage

```
mcmc_ds(Par, Data, cur.iter, adapt, Control, DM.hab.pois,
  DM.hab.bern = NULL, DM.det, Q, Prior.pars, Meta)
```

Arguments

`Par`	A list comprised of the following parameters: "det": a vector giving the current iteration's linear model parameters for the detection model; "hab.pois": a vector giving the current iteration's linear model parameters for Poisson abundance intensity; each row gives parameters for a particular species "hab.bern": a vector giving the current iteration's linear model parameters for Bernoulli part of ZIP model for abundance (if Meta$ZIP=TRUE) "cor": a correlation parameter for detections that's an increasing function of distance (correlation at the maximum distance); "Nu": a vector giving the log of the abundance intensity for each strata; "Eta.pois": If Meta$spat.ind==FALSE, spatial random effects for Poisson abundance model; one for each cell and for each species "Eta.bern": If Meta$spat.ind==FALSE & Meta$ZIP=TRUE, spatial random effects for Bernoulli abundance model; one for each cell and for each species "tau.eta.pois": If Meta$spat.ind==FALSE, precision for spatial ICAR model(s) for the Poisson component "tau.eta.bern": If Meta$spat.ind==FALSE & Meta$ZIP=TRUE, precision for spatial ICAR model(s) for the Bernoulli component "tau.nu": Precision for Nu (overdispersion relative to the Poisson distribution) "G": a vector giving the number of groups of animals in each strata; "N": a vector giving the number of animals in each strata "MisID": a list, each entry i of which is a vector holding the parameters for the ith misID model "Cov.par": an (n.species X n X n.ind.cov) array holding parameters of individual covariate distributions.
`Data`	A four dimensional array; the first dimension gives species, the second gives the transect, the third indexes a (possible) observation, and the fourth dimension gives observations and covariates associated with a given animal. These final columns are: Observer ID,Y(observation=0/1),Observed species,Obs covariates,Distance,Ind covariates
`cur.iter`	Number of iterations to run
`adapt`	If adapt==TRUE, run MCMC in adapt mode, optimizing MCMC proposal distributions prior to primary MCMC
`Control`	A list object including the following objects: "iter": number of MCMC iterations; "burnin": number of MCMC burnin iterations; "thin": if specified, how many iterations to skip between recorded posterior samples; "adapt": if adapt==TRUE, this gives the number of additional MCMC iterations should be performed to adapt MCMC proposals to optimal ranges prior to final MCMC run; "MH.cor": Metropolis-hastings tuning parameter for updating the correlation parameter (if Meta$point.ind==TRUE); "MH.nu": MH tuning parameter for Nu parameters (Langevin-Hastings multivariate update); "RJ.N": A vector giving the maximum number of additions and deletions proposed in an iteration of the RJMCMC algorithm for each transect "iter.fix.N" Number of iterations to skip RJMCMC step
`DM.hab.pois`	A design matrix for the Poisson model for abundance intensity (log scale)
`DM.hab.bern`	If Meta$ZIP=TRUE, a design matrix for the Bernoulli zero model (probit scale)
`DM.det`	A design matrix for the probit of detection probability
`Q`	An inverse precision matrix for the spatial ICAR process
`Prior.pars`	A list object giving parameters of prior distribution. Includes the following objects "a.eta": alpha parameter for prior precision of spatial process (assumed

	Gamma(a.eta,b.eta)) "b.eta": beta parameter for prior precision of spatial process (assumed Gamma(a.eta,b.eta)) "a.nu": alpha parameter for prior precision of overdispersion process (assumed Gamma(a.nu,b.nu)) "b.nu": beta parameter for prior precision of overdispersion process (assumed Gamma(a.nu,b.nu)) "beta.tau": Prior precision for regression coefficients "misID.mu": a list vector, each entry gives normal prior means for misID regression coefficients for the corresponding model in Meta$misID.mat (can be set to null if no misID) "misID.sd": a list vector, each entry gives normal prior sd for misID regression coefficients for the corresponding model in Meta$misID.mat (can be set to null if no misID)
`Meta`	A list object giving a number of other features of the dataset, including: "n.transects" Number of transects "n.species" Number of species "S" Number of strata cells "spat.ind" Indicator for spatial dependence "Area.hab" Vector giving relative area covered by each strata "Area.trans" Vector giving fraction of area of relevant strata covered by each transect "Adj" Adjacency matrix giving connectivity of spatial grid cells "Mapping" Vector mapping each transect into a parent strata "Covered.area" Vector giving the fraction of each strata covered by transects "n.Observers" Vector giving the number of observers that operated on each transect "M" Matrix with species-specific rows giving maximum possible value for number of groups present in each transect (in practice just set high enough that values at M and above are never sampled during MCMC) and can be fine tuned as needed#' "stacked.names" Character vector giving column names for the dataset "factor.ind" Indicator vector specifying whether data columns are factors (1) or continuous (0) "detect" If TRUE, detection parameters are estimated; if FALSE assumes a census "Det.formula" a formula object specifying the model for the detection process "Levels" a list object, whose elements are comprised of detection model names; each element gives total # of levels in the combined dataset "i.binned" indicator for whether distances are recorded in bins (1) or are continuous (0) "dist.pl" gives the column in Data where distances are located "G.transect" vector holding current number of groups of animals present in area covered by each transect "N.transect" vector holding current number of animals present in covered area by each transect "grps" indicator for whether observations are for groups rather than individuals "n.bins" number of distance bins (provided i.binned=1) "Bin.length" vector giving relative size of distance bins "n.ind.cov" Number of individual covariates (distance is not included in this total, but group size is) "Cov.prior.pdf" character vector giving the probability density function associated with each individual covariate (type ? hierarchical_DS for more info) "Cov.prior.parms" An (n.species X n X n.ind.cov) array providing "pseudo-prior" parameters for individual covarate distributions (only the first row used if a signle parameter distribution) "Cov.prior.fixed" indicator vector for whether parameters of each covariate distribution should be fixed within estimation routine "Cov.prior.n" (#species X #covariates) Matrix giving number of parameters in each covariate pdf "ZIP" If TRUE, fit a ZIP model to abundance "point.ind" Indicator for whether point independence assumed (if no, then no correlation modeled b/w multiple observers as function of distance) "last.ind" If TRUE (and point.ind=TRUE), point independence operates by assuming 0 dependence at the farthest bin "cor.const" If TRUE, forces estimates of correlation associated with point independence to be positive if last.ind==FALSE or neg-

ative if last.ind==TRUE (default is FALSE) "fix.tau.nu" Indicator for whether tau.nu should be fixed (1) or estimated(0) "srr" Indicator for whether a spatially restricted regression model should be employed (1) or not (0) "srr.tol" Threshold eigenvalue level for SRR; only eigenvectors with higher eigenvalues than srr.tol are included in SRR formulation "misID" If TRUE, misidentification of species is modeled "misID.mat" With true state on rows and assigned state on column, each positive entry provides an index to misID.models (i.e. what model to assume on multinomial logit space); a 0 indicates an impossible assigment; a negative number designates which column is to be obtained via subtraction "misID.models" A formula vector providing linar model-type formulas for each positive value of misID.mat. "misID.symm" If TRUE, classification probabilities assumed to be symmetric (e.g. pi^2|1=pi^1|2) "N.par.misID" A vector specifying the number of parameters needed for each misID model "N.hab.pois.par" A vector specifying the number of parameters needed for each species' Poisson abundance model "N.hab.bern.par" If fitting a ZIP model, this vector specifying the number of parameters needed for each species' Bernoulli zero model "post.loss" If TRUE, observed and predicted detections are compiled for posterior predictive loss

Value

returns a list with the following objects: "MCMC": An 'mcmc' object (see 'coda' R package) containing posterior samples; "Accept": A list object indicating the number of proposals that were accepted for parameters updated via Metropolis- or Langevin-Hastings algorithms; "Control": A list object giving MCMC tuning parameters (which are updated if the 'adapt' alorithm is used) "Obs.N": Records latent abundance in each transect; dimension is (n.species X # samples X # transects) "Pred.N": Posterior predictive distribution for abundance in each transect; obtained by sampling a Poisson distribution given current parameter values (with possible zero inflation) "Post": Holds posterior samples for strata specific group sizes ("Post$G") and abundance ("Post$N") "Obs.det": if Meta$post.loss=TRUE, an array holding observed detection types for posterior predictive loss calculations dim = c(n.transects,n.obs.types,n.obs.types) "Pred.det": if Meta$post.loss=TRUE, an array holding predicted detection types for posterior predictive loss calculations dim = c(n.mcmc.iter,n.transects,n.obs.types,n.obs.types)

Author(s)

Paul B. Conn

`plot_N_map` *function to plot a map of abundance. this was developed for spatio-temporal models in mind*

Description

function to plot a map of abundance. this was developed for spatio-temporal models in mind

Usage

```
plot_N_map(cur.t, N, Grid, highlight = NULL)
```

Arguments

`cur.t`	time step to plot
`N`	A vector of values to plot (e.g. abundance)
`Grid`	A list of SpatialPolygonsDataFrame (one for each time step) - holding survey unit spatial information
`highlight`	If provided, the rows of Grid[[cur.t]] to specially highlight

Value

A ggplot2 object

Author(s)

Paul B. Conn

`plot_obs_pred` *plot 'observed' versus predicted values for abundance of each species at each transect*

Description

plot 'observed' versus predicted values for abundance of each species at each transect

Usage

```
plot_obs_pred(Out)
```

Arguments

`Out`	Output list from "mcmc_ds.R"

Author(s)

Paul B. Conn

`post_loss` *function to calculate posterior predictive loss given the output object from hierarchicalDS*

Description

function to calculate posterior predictive loss given the output object from hierarchicalDS

Usage

```
post_loss(Out, burnin = 0)
```

Arguments

`Out`	Output object from running hierarchicalDS
`burnin`	Any additional #'s of values from beginning of chain to discard before calculating PPL statistic (default is 0)

Value

A matrix with posterior variance (P), sums of squares (G) for the posterior mean and median predictions (compared to Observations), and total posterior loss (D)

Author(s)

Paul B. Conn

`probit.fct` *Mrds probit detection and related functions*

Description

For independent observers, probit.fct computes observer-specific detection functions, conditional detection functions, delta dependence function, duplicate detection function (seen by both), and pooled detection function (seen by at least one).

Usage

```
probit.fct(x, formula, beta, rho, ...)
```

Arguments

`x`	vector of perpendicular distances
`formula`	linear probit formula for detection using distance and other covariates
`beta`	parameter values
`rho`	maximum correlation at largest distance
`...`	any number of named vectors of covariates used in the formula

Details

The vectors of covariate values can be of different lengths because expand.grid is used to create a dataframe of all unique combinations of the distances and covariate values and the detection and related values are computed for each combination. The covariate vector observer=1:2 is automatically included. The folowing is too long for the examples section: test=probit.fct(0:10,~distance,c(1,-.15),.8,size=1:3) par(mfrow=c(1,2)) with(test[test$observer==1,], plot(distance,p,ylim=c(0,1),xlab="Distance",ylab="Detect probability") points(distance,pc,pch=2) points(distance,dup,pch=3) points(distance,pool,pch=4) legend(1,.2,legend=c("Detection","Conditional detection","Duplicate detection","Pooled detection"),pch=1:4,bty="n") plot(distance,delta,xlab="Distance",ylab="Dependence"))

Value

dat dataframe with distance, observer, any covariates specified in ... and detection probability p, conditional detection probability pc, dupiicate detection dup, pooled detection pool and dependence pc/p=delta.

Author(s)

Jeff Laake

`rect_adj` *Produce an RW1 adjacency matrix for a rectangular grid for use with areal spatial models (queens move)*

Description

Produce an RW1 adjacency matrix for a rectangular grid for use with areal spatial models (queens move)

Usage

```
rect_adj(x, y, byrow = FALSE)
```

Arguments

`x`	number of cells on horizontal side of grid
`y`	number of cells on vertical side of grid
`byrow`	If TRUE, cell indices are filled along rows (default is FALSE)

Value

adjacency matrix

Author(s)

Paul Conn <`paul.conn@noaa.gov`>

`rect_adj_RW2` *Produce an RW2 Adjacency matrix for a rectangular grid for use with areal spatial models. This formulation uses cofficients inspired by a thin plate spline, as described in Rue & Held, section 3.4.2 Here I'm outputting an adjacency matrix of 'neighbor weights' which makes Q construction for regular latices easy to do when not trying to make inference about all cells (i.e., one can just eliminate rows and columns associated with cells one isn't interested in and set Q=-Adj+Diag(sum(Adj))*

Description

Produce an RW2 Adjacency matrix for a rectangular grid for use with areal spatial models. This formulation uses cofficients inspired by a thin plate spline, as described in Rue & Held, section 3.4.2 Here I'm outputting an adjacency matrix of 'neighbor weights' which makes Q construction for regular latices easy to do when not trying to make inference about all cells (i.e., one can just eliminate rows and columns associated with cells one isn't interested in and set Q=-Adj+Diag(sum(Adj))

Usage

```
rect_adj_RW2(x, y, byrow = FALSE)
```

Arguments

`x`	number of cells on horizontal side of grid
`y`	number of cells on vertical side of grid
`byrow`	If TRUE, cell indices are filled along rows (default is FALSE)

Value

adjacency matrix

Author(s)

Paul Conn <`paul.conn@noaa.gov`>

`rrw` *SIMULATE AN ICAR PROCESS*

Description

SIMULATE AN ICAR PROCESS

Usage

```
rrw(Q)
```

Arguments

`Q`	Precision matrix for the ICAR process

Value

Spatial random effects

Author(s)

Devin Johnson

`simdata` *MCMC output from running example in Hierarchical DS*

Description

MCMC output from running example in Hierarchical DS

Author(s)

Paul Conn `<paul.conn@noaa.gov>`

`simulate_data` *function to simulate double observer spatial distance sampling data subject to possible zero inflation and species misidentification*

Description

function to simulate double observer spatial distance sampling data subject to possible zero inflation and species misidentification

Usage

```
simulate_data(S, Observers, ZIP = TRUE, misID = TRUE, tau.pois = 15,
  tau.bern = 20)
```

Arguments

`S`	number of spatial strata (a single transect is placed in each strata and assumed to cover the whole strata)
`Observers`	A (2 x S) matrix giving the observer identity for each transect
`ZIP`	If TRUE, simulate abundance using a zero-inflated Poisson model
`misID`	If TRUE, assume species misidentification
`tau.pois`	Precision of the ICAR process for Poisson abundance
`tau.bern`	Precision of the ICAR process for zero inflation

Value

a distance sampling dataset

Author(s)

Paul B. Conn

`sim_out` *MCMC output from running example in Hierarchical DS*

Description

MCMC output from running example in Hierarchical DS

Author(s)

Paul Conn <`paul.conn@noaa.gov`>

`square_adj` *Produce an adjacency matrix for a square grid*

Description

Produce an adjacency matrix for a square grid

Usage

```
square_adj(x)
```

Arguments

`x`	number of cells on side of grid

Value

adjacency matrix

Author(s)

Paul Conn

`stack_data` *function to stack data (going from three dimensional array to a two dimensional array including only "existing" animals*

Description

function to stack data (going from three dimensional array to a two dimensional array including only "existing" animals

Usage

```
stack_data(Data, Obs.transect, n.transects, stacked.names, factor.ind)
```

Arguments

`Data`	three-d dataset
`Obs.transect`	current number of observations of animals in each transect (vector)
`n.transects`	number of transects
`stacked.names`	column names for new stacked dataset
`factor.ind`	a vector of indicator variables (1 = factor/categorical variable, 0 = continuous variable)

Value

a stacked dataset

Author(s)

Paul B. Conn

`stack_data_misID` *function to stack data for midID updates (going from four dimensional array to a two dimensional array including observed groups*

Description

function to stack data for midID updates (going from four dimensional array to a two dimensional array including observed groups

Usage

```
stack_data_misID(Data, G.obs, g.tot.obs, n.Observers, n.transects,
  n.species, stacked.names, factor.ind)
```

Arguments

`Data`	4-d dataset
`G.obs`	matrix giving the total numer of groups observed at least once by species and transect
`g.tot.obs`	total number of observations for animals seen at least once
`n.Observers`	vector giving number of observers per transect
`n.transects`	number of transects
`n.species`	number of species
`stacked.names`	column names for new stacked dataset
`factor.ind`	a vector of indicator variables (1 = factor/categorical variable, 0 = continuous variable)

Value

a stacked dataset (in matrix form)

Author(s)

Paul B. Conn

`summary_N` *calculate parameter estimates and confidence intervals for various loss functions*

Description

calculate parameter estimates and confidence intervals for various loss functions

Usage

```
summary_N(Out)
```

Arguments

`Out`	Output list from "mcmc_ds.R"

Value

summary.N list vector, with the first list index indicating species

Author(s)

Paul B. Conn

---

switch_pdf *function to calculate the joint pdf for a sample of values from one of a number of pdfs*

Description

function to calculate the joint pdf for a sample of values from one of a number of pdfs

Usage

```
switch_pdf(x, pdf, cur.par, RE)
```

Arguments

`x`	values to be evaluated
`pdf`	probability density function (pois1, poisson, pois1_ln, poisson_ln, normal, multinom)
`cur.par`	a vector giving parameters for the specified distribution; only the first is used for single parameter distributions
`RE`	random effects, if present

Value

total log likelihood of points

Author(s)

Paul B. Conn

---

switch_sample *function to sample from a specified probability density function*

Description

function to sample from a specified probability density function

Usage

```
switch_sample(n, pdf, cur.par, RE)
```

Arguments

`n`	number of samples desired
`pdf`	probability density function (pois1, poisson, normal, unif.disc, unif.cont)
`cur.par`	a vector giving parameters for the specified distribution; only the first is used for single parameter distributions
`RE`	random effects, if present

Value

a vector of length n samples from the desired distribution

Author(s)

Paul B. Conn

`switch_sample_prior` *function to sample from hyperpriors of a specified probability density function; note that initial values for sigma of lognormal random effects are fixed to a small value (0.05) to prevent numerical errors*

Description

function to sample from hyperpriors of a specified probability density function; note that initial values for sigma of lognormal random effects are fixed to a small value (0.05) to prevent numerical errors

Usage

```
switch_sample_prior(pdf, cur.par)
```

Arguments

`pdf`	probability density function (pois1, poisson, normal, unif.disc, unif.cont)
`cur.par`	a vector giving parameters for the specified distribution; only the first is used for single parameter distributions

Value

a vector of length n samples from the desired distribution

Author(s)

Paul B. Conn

`table.mcmc` *function to export posterior summaries from an mcmc object to a table*

Description

function to export posterior summaries from an mcmc object to a table

Usage

```
## S3 method for class 'mcmc'
table(MCMC, file = NULL, type = ”csv”, a = 0.05)
```

Arguments

`MCMC`	An mcmc object with columns referencing different parameter types (column names are used for plotting labels)
`file`	A file name to ouput to (including path); if null (default), outputs to screen
`type`	What type of table to produce (either "csv" or "tex")
`a`	Value to use for credible intervals. For example, alpha=0.05 results in 95% credible intervals

Author(s)

Paul B. Conn

Index